DOCUMENT RESUME

ED 294 713 SE 049 042

AUTHOR Kaput, James J.; Pattison-Gordon, Laurie
TITLE A Concrete-to-Abstract Software Ramp: Environments for Learning Multiplication, Division and Intensive Quantity. Technical Report 87-8.
INSTITUTION Educational Technology Center, Cambridge, MA.
SPONS AGENCY Office of Educational Research and Improvement (ED), Washington, DC.
PUB DATE Sep 87
CONTRACT 400-83-0041
NOTE 51p.; Some drawings and charts may not reproduce well.
PUB TYPE Reports - Descriptive (141) -- Reports - Research/Technical (143)

EDRS PRICE MF01/PC03 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; Computer Uses in Education; Courseware; *Division; *Elementary School Mathematics; Elementary Secondary Education; *Mathematics Curriculum; Mathematics Education; Mathematics Instruction; *Multiplication; *Secondary School Mathematics

ABSTRACT

This document is intended to describe several software learning environments in an order that parallels a reasonable sequence of use by students. It also describes a planned an designed, but not yet implemented, extension of these environments from the discrete to the continuous case. Each of the implemented environments was developed in continuous collaboration with teachers and their students from grades three to eight. An important characteristic of these environments is their systematic linking of concrete representations, beginning with iconic representations, to more abstract representations. The report is designed to serve two functions: (1) as an overview to learning the conceptual field of multiplicative structures; and (2) as a guide to software environments. The remainder of the report is organized according to the design of the software environments. This includes descriptions of the icon-based calculation environments, the environment that introduces the numerical and graphic representations, the algebraic representations that extend these environments, the sampling environments, and finally the transitions to the continuous environment. (TW)

TR87-8

ED294713

A CONCRETE-TO-ABSTRACT SOFTWARE RAMP: ENVIRONMENTS FOR LEARNING MULTIPLICATION, DIVISION, AND INTENSIVE QUANTITY

Technical Report

September 1987

SE049042

Educational Technology Center
Harvard Graduate School of Education
337 Gutman Library Appian Way Cambridge MA 02138
(617) 495-9373

A Concrete-to-Abstract Software Ramp: Environments for Learning Multiplication, Division, and Intensive Quantity

Technical Report
September 1987

Prepared by:

James J. Kaput
Laurie Pattison-Gordon

Word Problems Project

Group Members

Kathy Hollowell
James J. Kaput
Mary Maxwell Katz
Joel Lubin
Clifton Luke
Laurie Pattison-Gordon
Joel Poholsky
Yolanda Rodriguez
Aline Sayer
Judah L. Schwartz
Jane West
Philip Zodhiates

Software Design and Programming

Laurie Pattison-Gordon

Preparation of this Report was supported in part by the Office of Educational Research and Improvement (Contract #OERI 400-83-0041). Opinions expressed herein are not necessarily shared by OERI and do not represent Office policy.

A Concrete-to-Abstract Software Ramp: Environments for Learning Multiplication, Division and Intensive Quantity

TABLE of CONTENTS

Acknowledgements

We would like to express our thanks to Joel Poholsky and Aline Sayer for their careful reading and insightful critique of this document. We extend our thanks to the teachers in the Word Problems Project making suggestions on the software and lesson design, and for testing and evaluting the software. Their comments have been invaluable. We also thank Laurel Katz for designing and programming the icons for the software.

A Concrete-to-Abstract Software Ramp: Environments for Learning Multiplication, Division and Intensive Quantity

INTRODUCTION

This document will describe the several software learning environments that our group has developed over the past two years, in an order that parallels a reasonable sequence of use by students. We will also describe a planned and designed, but not yet implemented, extension of these environments from the discrete to the continuous case.

Each of the implemented environments was developed in continuous collaboration with teachers and their students from grades 3 to 8. The story of the clinical and classroom work associated with this development will be available as a series of papers currently in preparation. See also (Kaput, et al, 1986; 1987) for additional empirical background.

As is described in much more detail in other reports on this Project (Kaput, 1985; Kaput, et al, 1986), a central objective of all our learning environments is to ramp students from their concrete, situation-bound thinking about the conceptual field of multiplicative structures to more abstract and flexible thinking. This conceptual field includes concepts beginning with multiplication and division and increasing in complexity through proportions all the way to vector spaces (Vergnaud, 1983). We are interested here in that portion of the conceptual field that appears in the core school curriculum up to the solution of proportions by simple algebraic equations.

An important, perhaps defining, characteristic of these environments is their systematic linking of concrete representations, beginning with iconic representations, to the more abstract representations. It is on this abstract representation that much of advanced mathematics is built, e.g., coordinate graphs and algebraic functions. This systematic linking of computer representations is the foundation of our strategy for building rich and flexible cognitive representations. It is an attempt to build meaning and understanding gradually in this conceptual field in a way that will support long term competence - computational and conceptual competence.

An immediate result of this long term view is the fact that these types of environments are intended to be used over a period of several years and grade levels. We do not believe in quick-fix technological solutions to difficult and long standing learning problems, and agree with Vergnaud in his view that important conceptual structures take many years to develop. To short-cut this extended meaning-building process, perhaps in the hopes of teaching near-term procedural performance, is to invite long term disaster, precisely the disaster that is now widely recognized as having occurred in so many nations across the world (McKnight, et al, 1987).

A second important characteristic of the learning environments is that they put students in situations calling for actions using a particular representation, where the students are then responsible for judging the adequacy or appropriateness of those actions, often by interpreting relative to a representation different from the one in which the action was initiated. The computer reports, the students evaluate, not the other way around!

A third characteristic of the learning environments is the assumption that all numbers involved refer to something - numbers are used in the context of quantity. We do not engage the students in abstract arithmetic. This is as much a property of the written materials and discussion associated with the use of the software as of the software itself, which requires that all numbers have units associated with them.

Some Practical Matters

The learning environments were designed as research tools and constitute prototype software. In their current form they are not intended for general classroom use except in situations involving teachers experienced in the use of beta-version software. Indeed, we regard them more as "pointers" to model uses of computers in this domain of mathematics than as examples of such. They are available for research purposes from the Center at nominal cost. They were written for a Macintosh in True BASIC, a Pascal-like programming environment which, unfortunately, in an attempt to be fully transportable across very different hardware and operating systems, does not facilitate full use of the Macintosh windowing and command environment. Our use of multiple representations linked across different windows and our application of icon-based calculation systems, in an ideal implementation, would be especially well served by that environment, because the user would be able to control the size, visibility, orientation and even choice of the different windows that are now frozen in place. The interface would also be less cluttered and more consistent. But, as indicated, this was specifically not a development project, but a learning research project.

Organization of the Report

We intend the Report to serve two functions, (1) as an overview of our approach to learning the conceptual field of multiplicative structures, and (2) as a guide to the software environments themselves. However, this document will not provide sufficient detail to serve as complete user documentation for the software itself, especially since the software is continually evolving as experience with it accumulates. Any potential user of these environments should obtain updates, especially since minor changes in certain details can lead to major frustrations for a user not aware of those changes. Furthermore, the software is used in conjunction with substantial amounts of accompanying written materials, which, although they will be alluded to from time to time, are not included here. Indeed, matters of pedagogy and curriculum are too important to be covered as an ancillary matter in a software description, so will be dealt with in separate documents. Scattered among the many screen-dumps that comprise our major display tool will be found rationales for the design decisions taken.

The remainder of the report will be organized according to the design of the software environments. Hence we will begin with descriptions of the icon-based calculation environments, then move to the environment that introduces the numerical and graphic representations, discuss the algebraic representations that extend these environments, introduce the sampling environments, and finally introduce the transitions to the continuous environment. Lastly we will discuss our designs of the continuous environment. Thus the bulk of the software is centered on introducing intensive quantities in the discrete case first, because the arithmetic is simplest, and because a reasonable conceptual distance can be maintained between the part-part relationships associated with discrete intensive quantities and the part-whole relationships that are at the basis of students' experience with fractions(Behr, et al, 1983).

AN OVERVIEW OF THE SOFTWARE

The existing software is organized into the top four environments indicated in Diagram 1, and the fifth, the continuous environment, is currently being implemented.

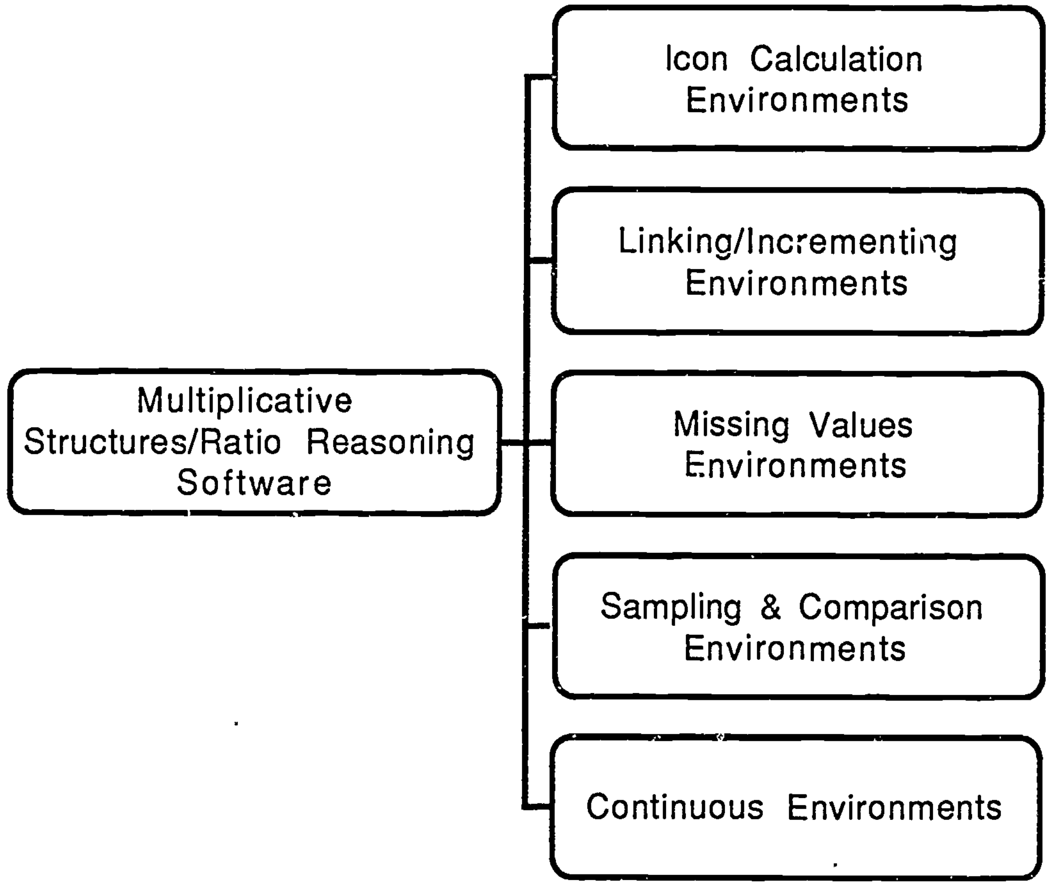

Diagram 1

The "software ramp" begins at the most concrete level with the icon calculation environments, of which there are two families, as indicated in Diagram 2.

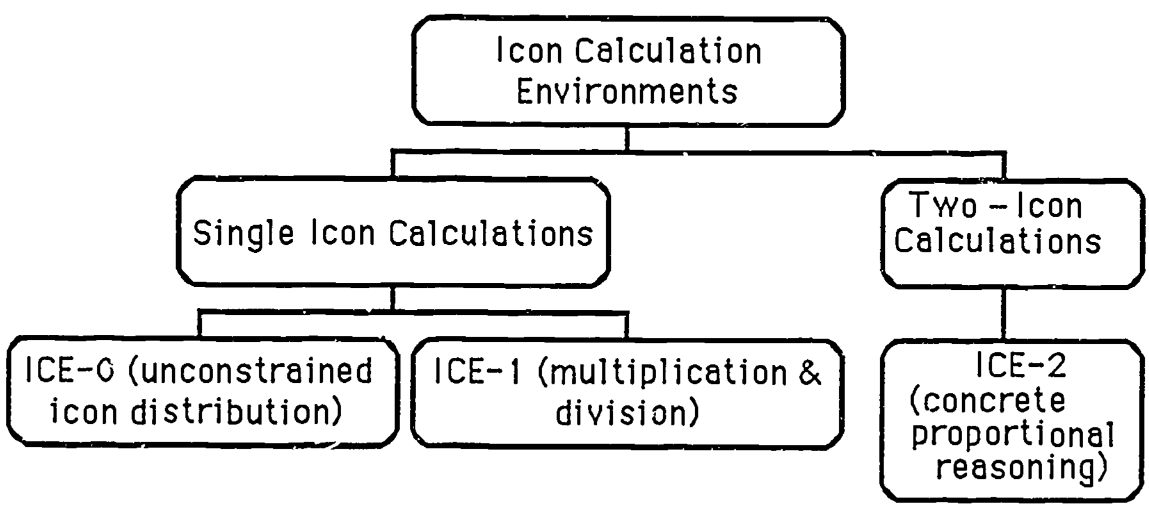

Diagram 2

One family, on the left of the diagram, supports calculations with single icon sets - multiplication and the two forms of division, although ICE-0 can be used as an organizer for simple calculations of any kind ("ICE" stands for Icon Calculation Environment."). It acts simply as a "game board" on which one can take icons from a reservoir and freely distribute them in a rectangular array of cells. In ICE-1, the computer provides elementary numerical information on one's actions, e.g., how many icons have been distributed, how many cells have been used, etc.

The second family supports calculations coordinating the distribution of <u>two</u> sets of icons, as occurs in concrete, iconic versions of proportion solving. In this context, proportional reasoning - historically assumed to be a difficult reasoning process requiring substantial intellectual development - can be accomplished using only the fundamental cognitive acts of grouping, matching and counting.

The process of ramping upward from these concrete reasoning environments to the more powerful mathematical representations such as coordinate graph and algebraic equations is achieved through the application of a set of Linking environments - see Diagram 3. These support simple acts of incrementing numbers where the student can examine at will the linked consequences of these incrementing (and decrementing) acts in any or all of four representations: Iconic, Numeric (tables of data), Graphic (coordinate graphs), and Algebraic (equations). There are also representations designed explicitly to facilitate the transition from the concrete to the abstract, for example the concrete transitional coordinate graph that stacks icons along the axes instead of simply labeling axes with numbers.

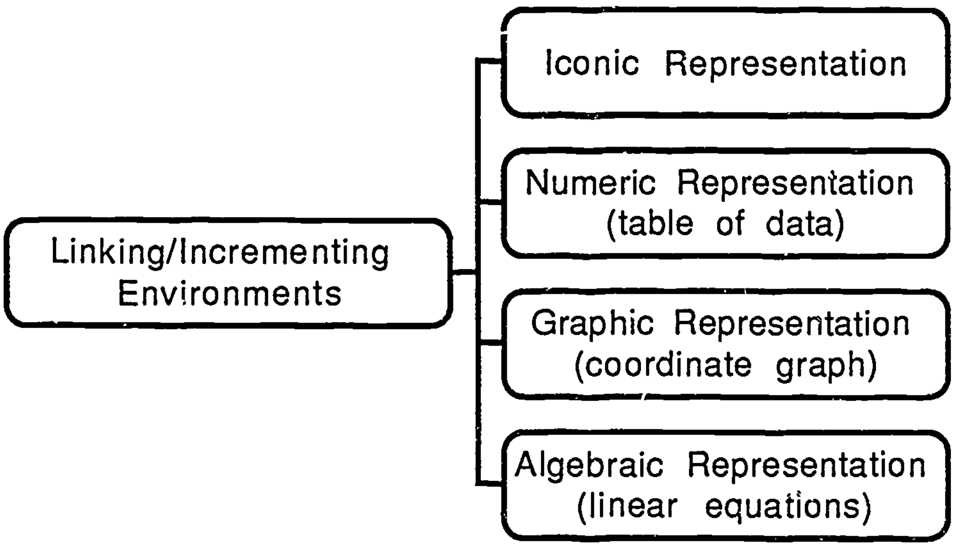

Diagram 3

Once new representations are available in which to do proportional reasoning, the student can then explore the different forms of that reasoning process across those representations. Recall that to solve "missing vaue problems" is to engage in the primary form of proportional reasoning: one is given a ratio in the form of a pair of numbers, and a third number. The task is to determine a fourth number that fits the given ratio. In the Missing Values Environments students can input solutions in any of four representations and check the results achieved in any of the anothers. This power is most often exercised when a student uses a more concrete representation (iconic or numeric) to check on computations done relative to a more abstract representation (coordinate graph or algebraic equation) - see Diagram 4.

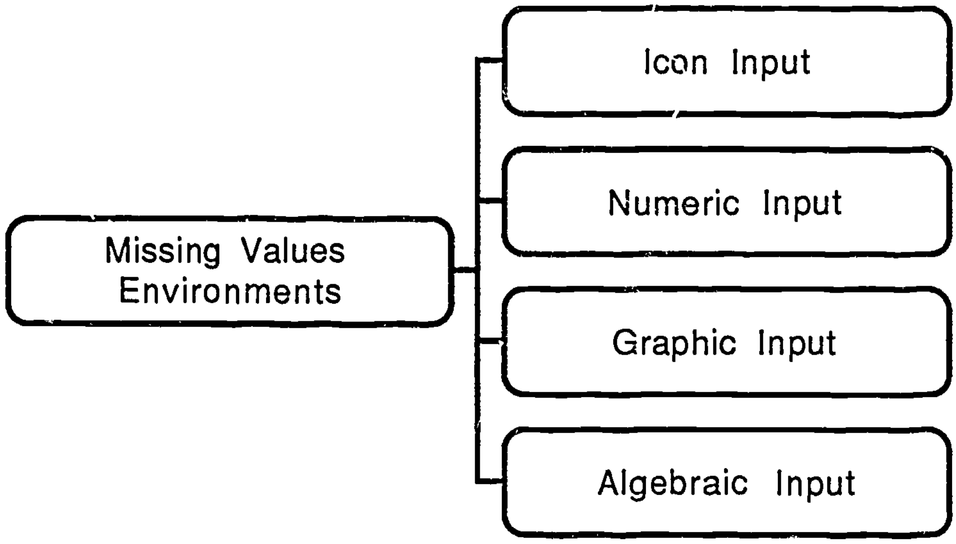

Diagram 4

Another aspect of ratios relates to their use in the describing of an internal feature or attribute of a substance, entity, or situation - for example: density in grams per cubic centemeter, speed of a car on a trip in miles per hour, or number of candies per child - these are sometimes referred to as "intensive quantities." This aspect of the complex idea is addressed in the various sampling environments - see Diagram 5 - where the matter of homogeneity of samples is also addressed, e.g., is the density constant?

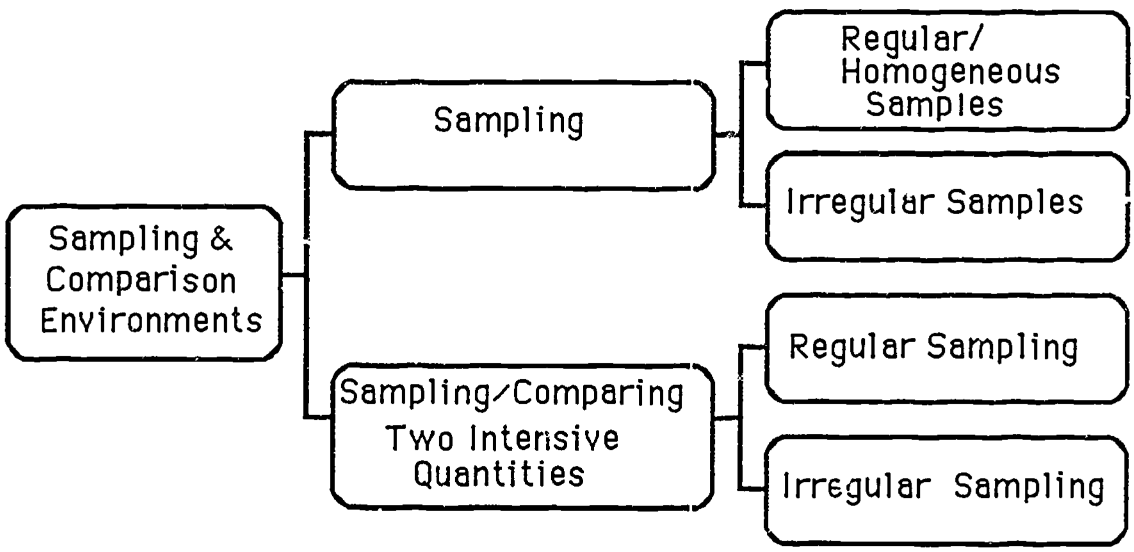

Diagram 5

These environments also support the comparison of intensive quantities. For example, if one park's picnic areas have 2 trees per 3 benches, and another has 3 trees per 5 benches, which is shadier? These environments thus engage yet a third aspect of this web of ideas, that of order.

But all of the previously mentioned software environments involved only discrete quantities, whereas the bulk of the mathematics curriculum beyond the early grades uses continuous quantities. Hence we have under development a series of linked representations where the numerosity of icons is replaced by length of continuous line segments. In addition to containing deliberate transitions from discrete to continuous representations, these environments also utilize

an interface and a command structure maximally parallel to that of the discrete world so as to induce transfer of understanding where appropriate and to minimize the difficulty of learning this more complex set of ideas - see Diagram 6.

The transition from the discrete to the continuous also provides an opportunity to test in a real curricular context the potential of deliberately manipulated interface and command structures to support important learning, especially transfer.

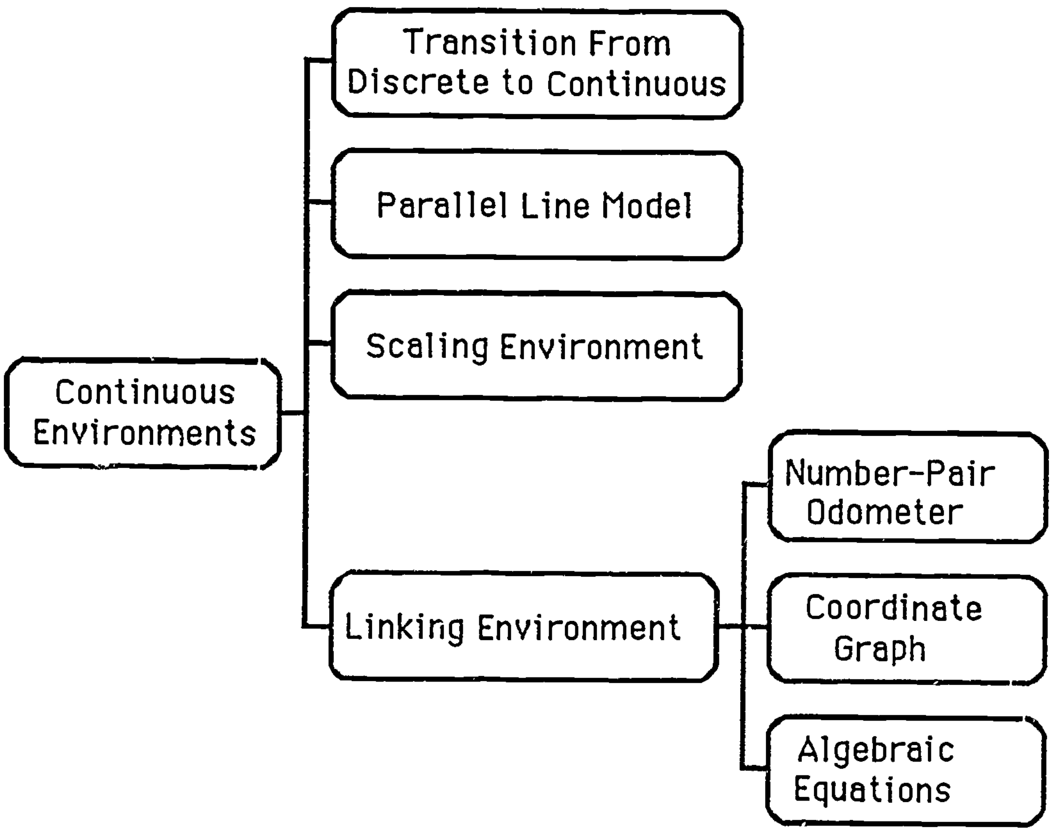

Diagram 6

1. INTRODUCING THE DISCRETE REPRESENTATIONS

Before systematically describing each of the environments, it will be useful to have an image of the different representations and a sense of why the sequence begins where it does, with an icon calculation environment that supports concrete multiplication and division.

In most of our software environments, the first screen the student is faced with is the menu of icons from which to choose - by pointing and clicking, as in Figure 1. (This set of icons could, of course, be enlarged straightforwardly, and in an ideal implementation, students would be able to sketch their own icons.)

In several of the environments involving a discrete intensive quantity, the next step is to build a model cell with two chosen icons by dragging replicas into a rectangular cell (which shrink upon sliding into the cell). In order to give a sense of the "intensivity" or intensive attribute of the collection, some of the environments flood the screen with copies of the model cell when the student signals completion of the cell building.

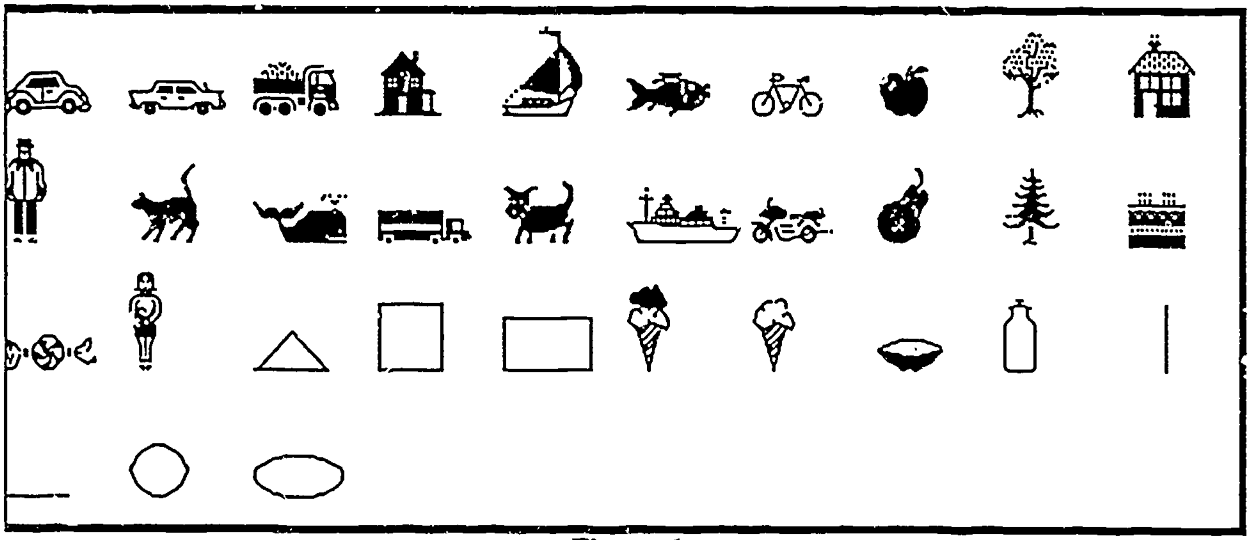

Figure 1

Note that the student can place the icons in the cell as he/she pleases. In particular, the icons can be arranged to reflect visually a semantic relationship among the entities represented. In the case pictured in Figure 2 below, the story-line assumption is that

We are planning to plant trees in a park so that every two trees will shade three people.

Hence the icons are laid out to reflect the trees-shading-people semantic relationship.

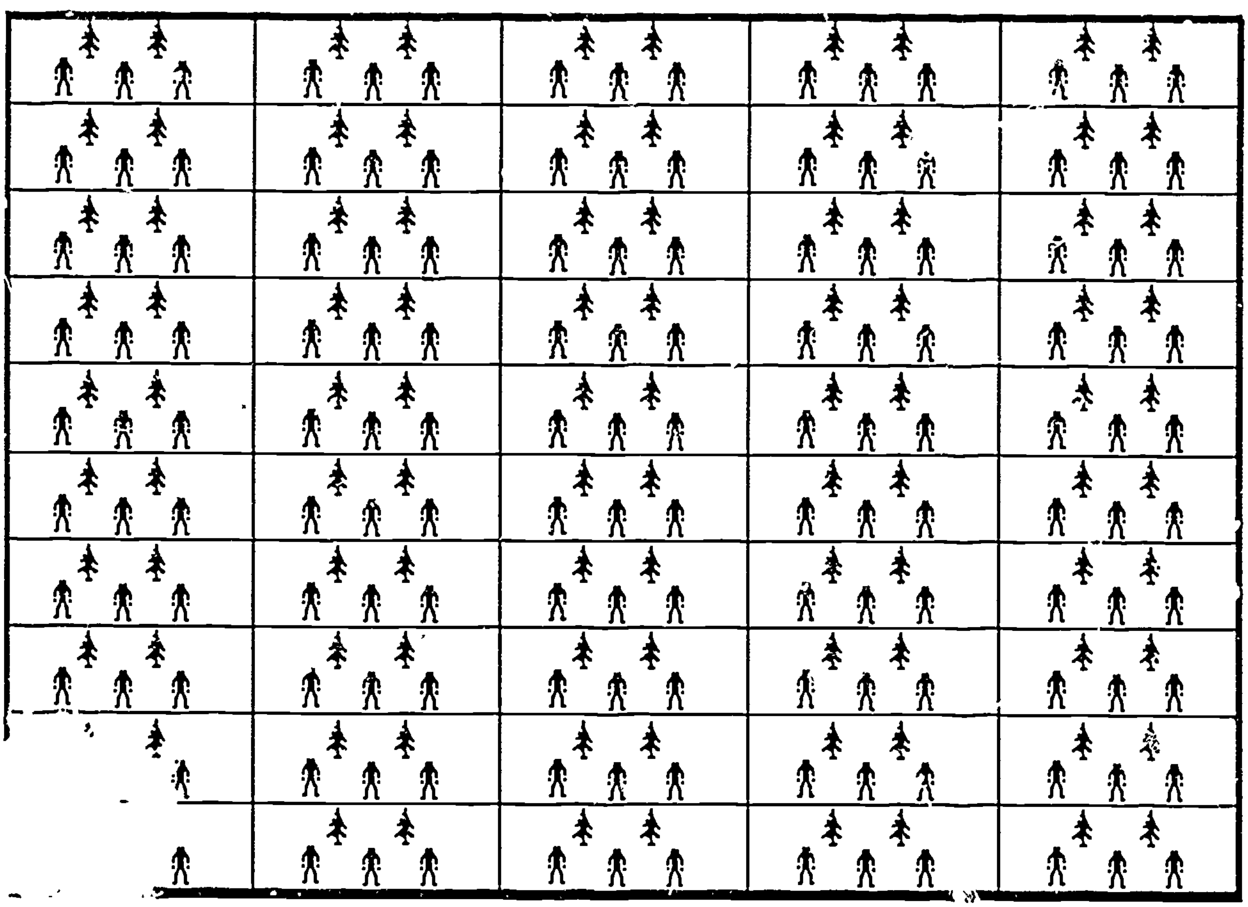

Figure 2

In one of the discrete environments, the "Linking Environment," it is possible to display all four representations simultaneously - the iconic, the numeric, the coordinate graphic, and the algebraic. This would seldom, if ever, be done in a school setting because the different representations involved might be introduced as many as two to four years apart.

In the environment shown below in Figure 3, all four representations are driven by the MORE and FEWER buttons, which increment or decrement the values of the variables in each representation as indicated (this particular screen would result after 5 clicks on MORE).

Typically, to control the amount of information on the screen several of the screen windows are toggled off (by clicking on their boundaries) so that only one or two are visible at a time. Indeed, this is a general convention in all the software environments:

A wide vertical boundary of a representation's window serves as a two-way toggle switch, where the two values are Visible and Invisibie (blank). When the window is visible, clicking on a wide horizontal boundary toggles between Active and Inactive.

This rule does not apply, of course, to those representations in which the actions of the environment are initiated.

In Figure 3 one can also choose which equation(s) to have active in this particular algebra window by clicking on the shaded area between expressions to turn on that particular equation (note the box in the lower right hand corner of the equation window). In the screen shown, all but one are active where the values of the variables appear as highlighted numbers in rectangular "frames" on the icons - matching the highlighted numbers in the table of data, of course. The given numbers of the initial ratio (2 and 3 in this case) are not highlighted.

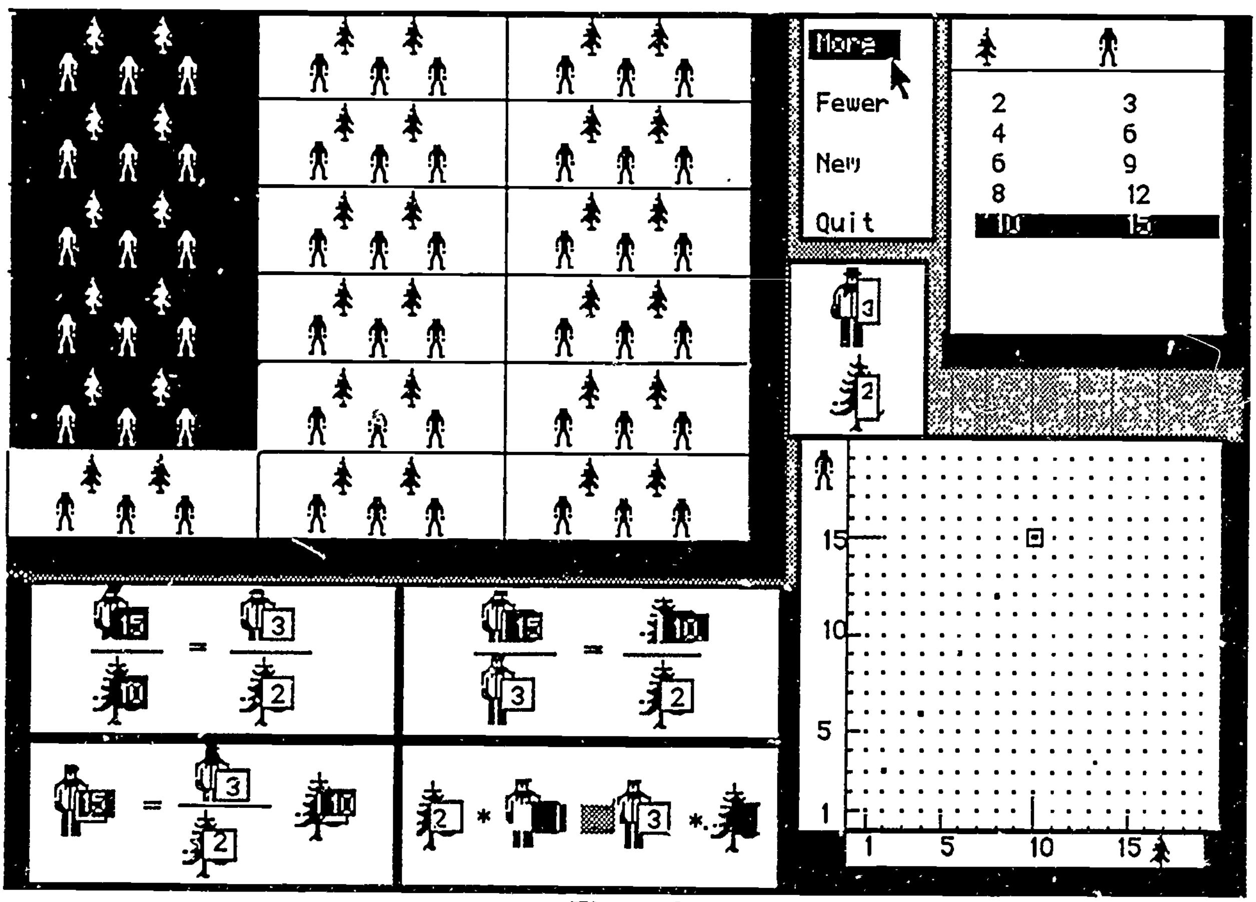

Figure 3

As a result, the intensive quantity is modeled in the coordinate graph as the slope of a line of discrete points.

1.1 The Boxes Strategy

In order to motivate the description of the icon-based environments intended to be used prior to the introduction of other representations, we recount a strategy spontaneously generated by a significant number of sixth and seventh grade students when solving missing value problems, problems such as:

How many people would be shaded by 14 trees in our park?

We observed students who had prior experience with the rectangular cell layout of the iconic representation, in either paper-pencil or computer based activities. We refer to their approach as the "boxes strategy." It is directly based on the rectangular cell organization of the iconic representation of the intensive quantity "2 trees per 3 people."

"Let's see, there are 2 trees per box and so there are 7 boxes of trees. There will be 7 times 3, that's 21 people."

This is a divide-and-then-multiply strategy based on an intermediate decomposition of the sets of icons into subsets describable using the intensive quantities

2 trees/box and 3 people/box.

These subsets then encourage a particular sequence of computations. The first is a quotative (or "measurement") division - a division of an extensive quantity by an intensive quantity: divide the given number of trees by the number of trees/box to get the number of boxes. Then multiply this number of boxes by the number of people/box to get the required number of people - the product of an intensive and an extensive quantity.

The component multiplication and division operations of this strategy are instantiated in the single-icon environment that is the first to be described in our sequence.

2. CONCRETELY BASED MULTIPLICATION AND DIVISION: ICE-0 and ICE-1.

Icon Calculation Environments

- Single Icon Calculations
  - ICE-0 (unconstrained icon distribution)
  - ICE-1 (multiplication & division)
- Two Icon Calculations
  - ICE-2 (concrete proportional reasoning)

Chart 1

After working more than a year with environments that used more than one representation to represent ratios and which support the solving of traditional missing value problems, it became clear that students needed more experience with the fundamental operations of multiplication and

division. In particular, it was evident that students need to make conceptual distinctions between the two types of division in order to make competent choices of operation to use in different situations, and to provide complementary structures to the different meanings of multiplication that occur in models involving intensive quantities. Hence we stepped backwards towards what we feel is the conceptual bedrock that will support viable thinking patterns for the longer run.

We devised a series of Icon Calculation Environments (hence the acronym "ICE") beginning with a single-icon calculation environment that utilizes primitive acts such as grouping, counting and matching, on a single set of icons.These acts are active concrete versions of multiplication of intensive by extensive quantities, and partitive and quotative division.

2.1 The Simple Object-Manipulation Environment: ICE-0

The student picks an icon to match some story situation. Let's suppose we are working on planning for the planting of trees in the park situation as described above. Hence we pick a tree icon from the menu in Figure 1. But rather than build a model cell as in Figure 2, we are presented with a screen as in Figure 4 below. This is as close we come to a pure, non-numerical object-manipulation environment, where we have a reservoir of icons to grab and drag into the cells.

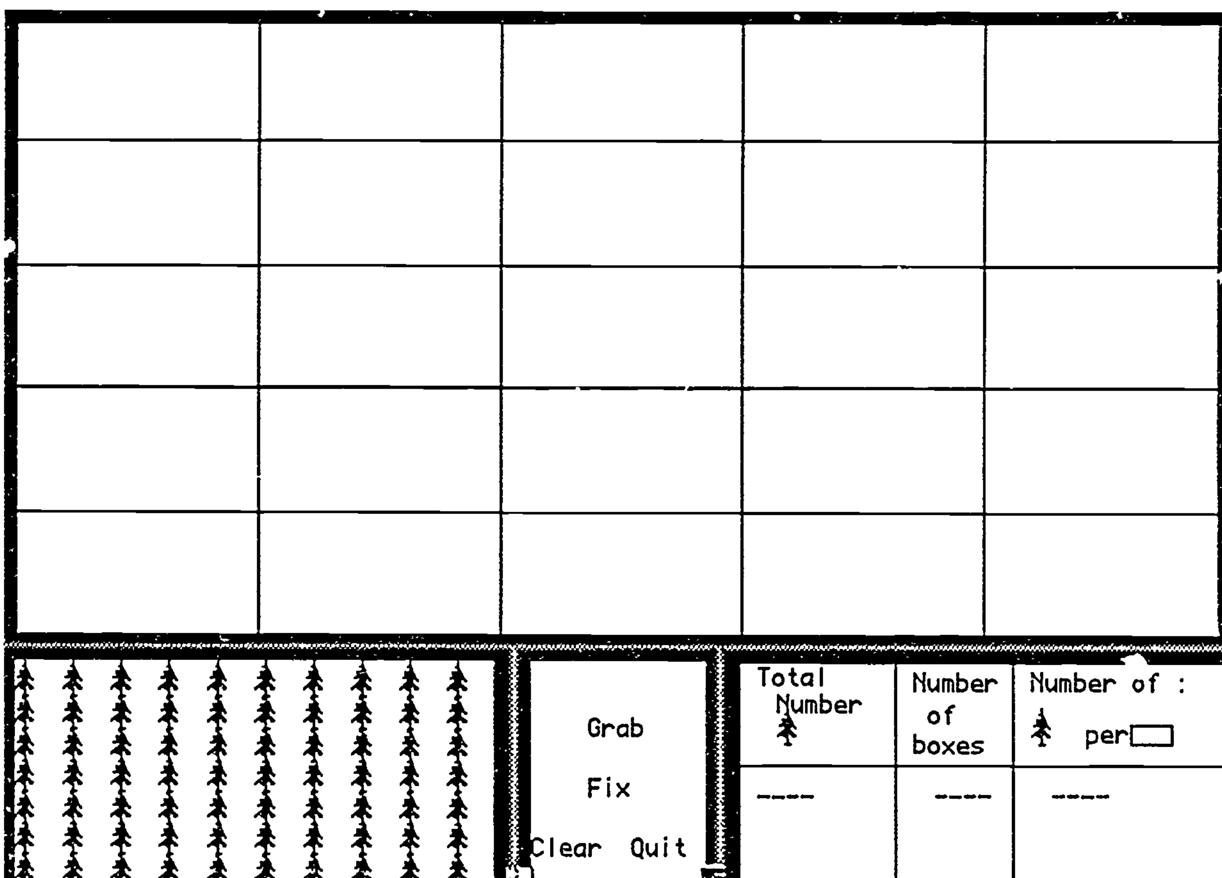

Figure 4

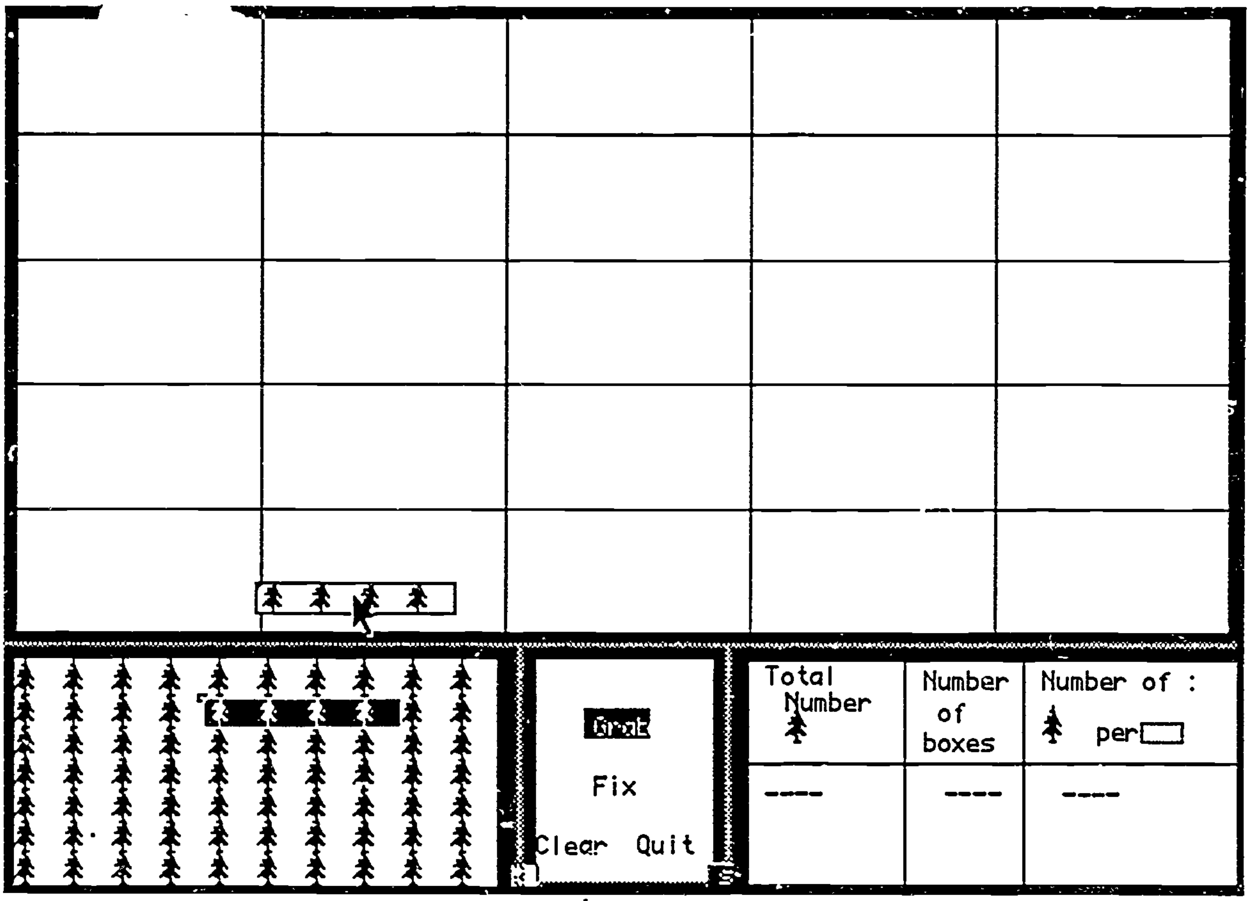

Figure 5

We signal that we want to "grab" by clicking on that command and then choose the icons to be dragged up into the cells by clicking on a point in the reservoir and then stretching an elastic rectangle over as many icons as we wish to move by holding the mouse button down as we stretch the rectangle. Releasing the button fixes the set to be moved. Then by pointing to the set of selected icons, holding the mouse button down and dragging the pointer upward, the set of icons can be dragged up into any of the cells above, as in Figure 5.

A double click in the destination cell deposits the icons in that cell. In the reservoir, the absence of that set of icons is represented by gray inverse video shadows in their former positions.

By clicking on the boundary of the Data window in the lower right hand corner, this window becomes active, displaying the number of tree icons moved, the number of boxes (cells) occupied with icons, and the number of icons per box if the number is the same for all occupied boxes. If not, that portion of the Data window says "number in boxes is not the same." In Figure 6 the data concerns only the one box we have dropped icons into, so matching boxes is moot. (Another version of this environment fills in the Data window automatically.)

It is also possible to use this free object-manipulation environment with the data table entirely suppressed by clicking on its left side boundary. This can be useful to create a mind-set that encourages thinking with icons rather than jumping prematurely to arithmetic and the manipulation of numerals.

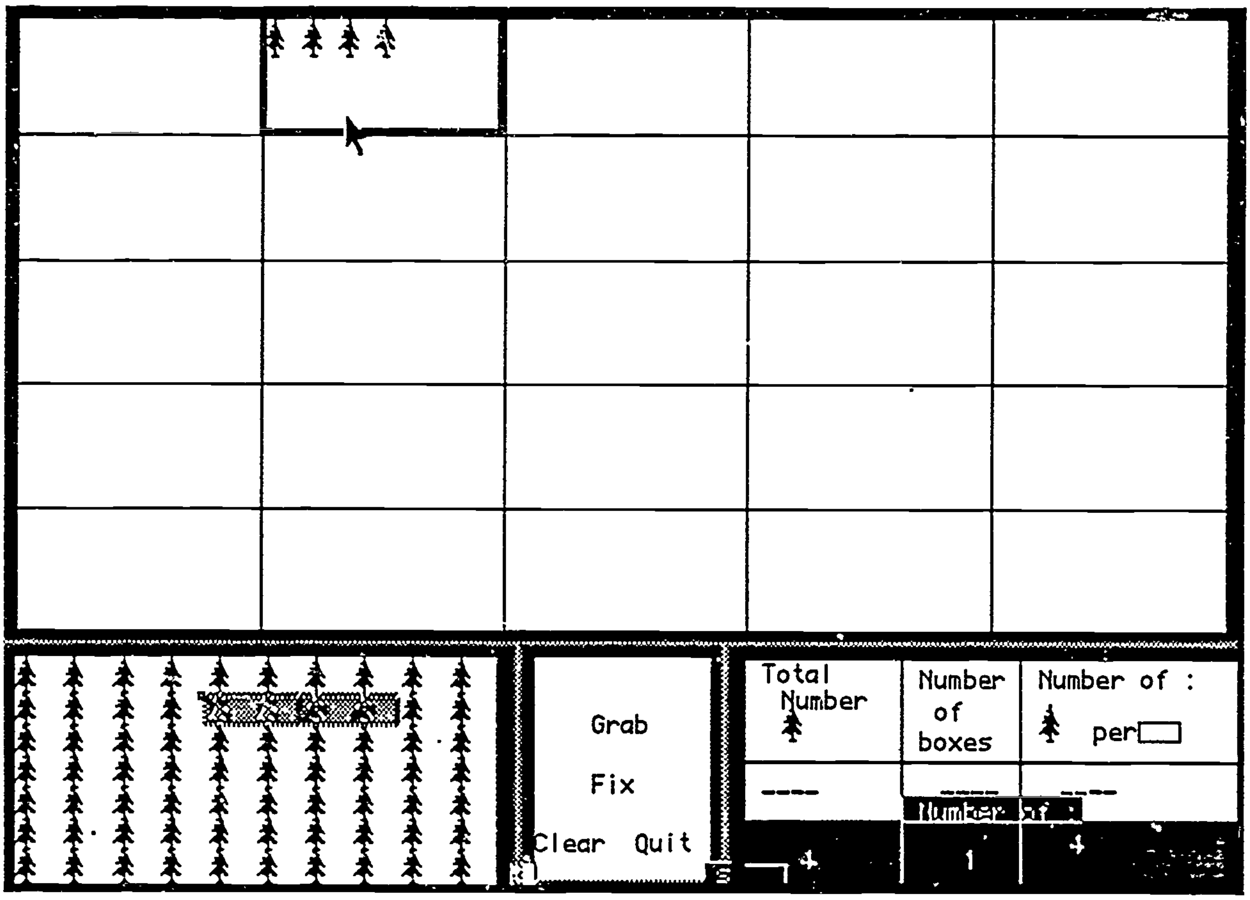

Figure 6

It is clear that the only structural difference between this computer environment and common classroom physical chip trading is the reporting or feedback provided by the computer. The other differences have to do with the orderliness of the transactions and the stability of the results - if one were to move several sets of icons, there is a plain record of what has been moved and where it has been moved to. A cybernetic manipulation environment is also more acceptable to older children than physical manipulatives, which are often regarded as "for babies."

By clicking on the Fix command, one is able to select and move any icons from any cell into any other or back into the reservoir.

Note that even this kind of an environment can be used to engage students in complex problem solving tasks that use the icons as indices for counting or distribution activities. For example, one could model the number of passengers on a train after repeated stops, during which passengers get on and off as well as switch cars. Such a task is, of course, primarily an additive exercise. Other, more combinatoric problems can also be modeled in this kind of environment.

2.2 Multiplication and the Two Divisions: ICE-1.

This environment provides concrete means for solving the equation E' = E*I for any of its quantity-variables when given the other two. Hence there are three problem types possible, one for each possible missing value in the equation. We have used the letters E and E' to denote extensive quantities and I to denote an intensive ("per") quantity. The quantity E' is instantiated as the number of icons, E as the number of boxes, and I as the number of icons per box. Thus, with reference to the data window in the lower right corner of Figure 6 above, providing the three respective problem types amounts to providing numbers in two of the three positions of that window and requesting the third. In particular,

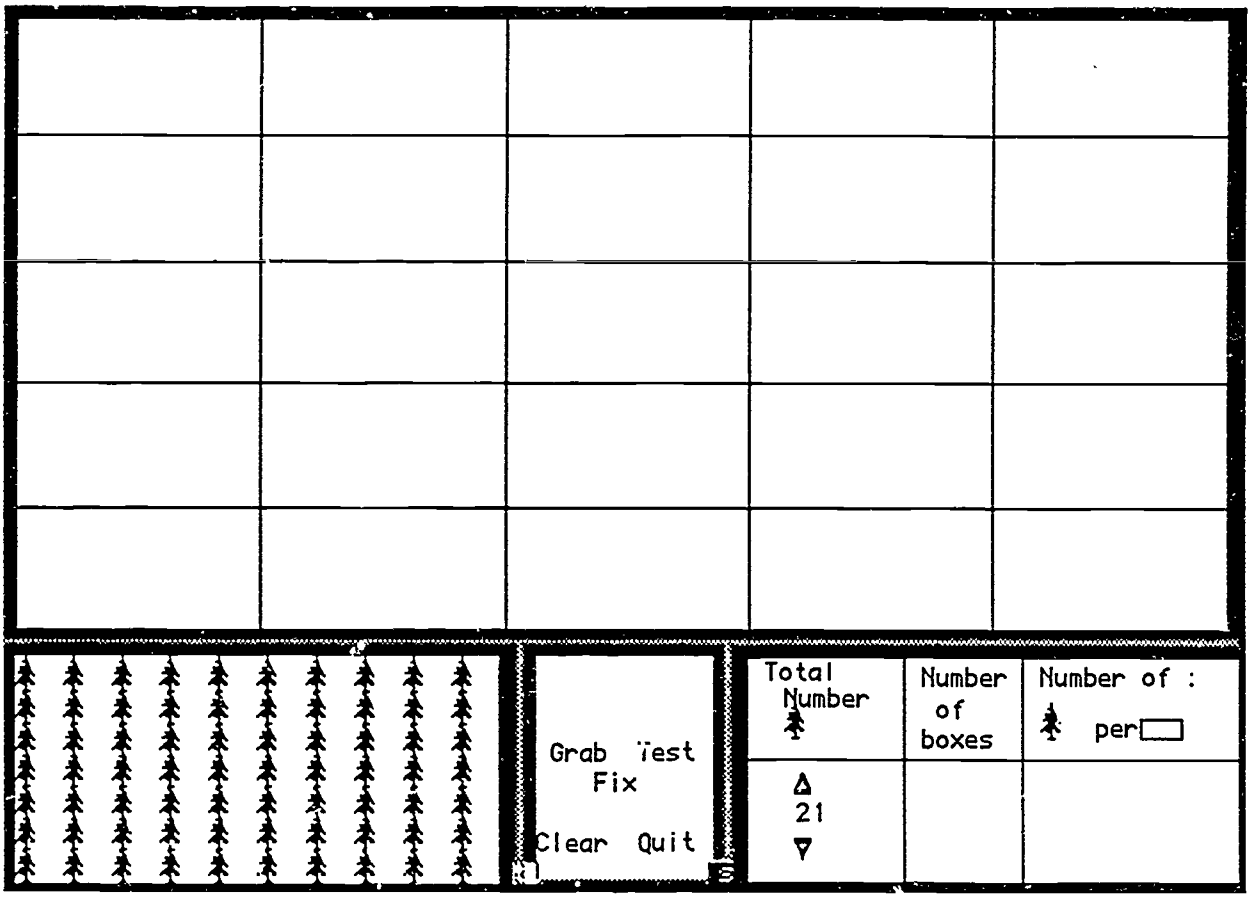

Figure 7

(1) providing the number of boxes and the number of icons/box yields a "rate" type of multiplication problem (find E*I: the total number of icons);

(2) providing the number of boxes and the total number of icons yields a partitive (fair share) division problem (find E'/E: the number of icons/box);

(3) providing the number of icons/box and the total number of icons yields a quotative division problem (find E'/I: the total number of boxes).

Total Number Items	Total Number of Boxes	Items per Box
given	given	**? - partitive division**
? - multiplication	given	given
given	**? - quotative division**	given

After picking an icon from the menu, the user is asked whether (1) the computer should generate the problems (both the type and the values), (2) whether the user should generate the problem type and the computer generate the numbers, or (3) whether the user should generate both the type and the numbers. If the user chooses to generate problem type only, the user must click on the quantities in the data window which are to be provided by the computer. The computer will randomly generate problem numbers amenable to solution using the numbers of boxes and icons available, and its division problems do not yield remainders. Whatever the choice, a screen appears in the same style as in Figure 6 above.

Let us now assume that the user has chosen to input both the problems and the numbers, so a screen such as that in Figure 7 appears - but with no numbers appearing in the data or "problem" window. (Its function changes from a problem set-up or statement window at the beginning, to a feedback window later, in response to the Test command.)

The user chooses the quantity to be input by clicking on the appropriate quantity label-area in the problem window. Then two arrowheads, separated by a zero, appear in the previously empty space below that label. For each click on an arrowhead, the number increments by one if it is the "up" arrow, or decrements if it is the "down" arrow. When the desired number is reached, the user clicks on that number itself to fix it in place before moving on to input the second quantity.

Alternatively, the user can input using the keyboard by first hitting return, then typing in the desired number after the "?" appears, and hitting return again.

After choosing two values, the user clicks in the lower left corner of the problem window.

Here in Figure 8 we have input the total number of trees and the number of boxes, so we have defined a fair share (partitive) division problem. We thus need to determine how many trees each of the 3 boxes will get in an equal distribution of the 21 trees. This problem can be solved by grabbing and dragging tree-icons into the cells above, as described earlier.

However, after clicking to indicate the problem has been satisfactorily set, the user is offered help, as indicated by the following message which will appear in the problem window, as shown in Figure 8.

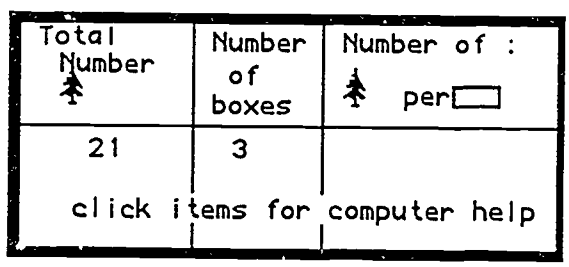

Figure 8

This important choice option determines the specific type of computer help, or scaffolding, that will be provided for the user's actions in solving the problem. The essence of this help option is to provide support for the component cognitive operations needed to solve the problem in this representation. Depending on the given quantity that the user clicks on, the help will differ, and no response will occur if the user clicks on the quantity to be determined. If the user clicks on the Total Number quantity, then exactly the correct number of icons will be made available in the reservoir and all the others will be deleted, as indicated in Figure 9. This helps the user by eliminating the need to keep a running count of the total number of icons taken from the reservoir - one is finished when the supply is exhausted. This becomes a significant help in the face of the uncertainty that occurs in students' early experience with problems having a remainder.

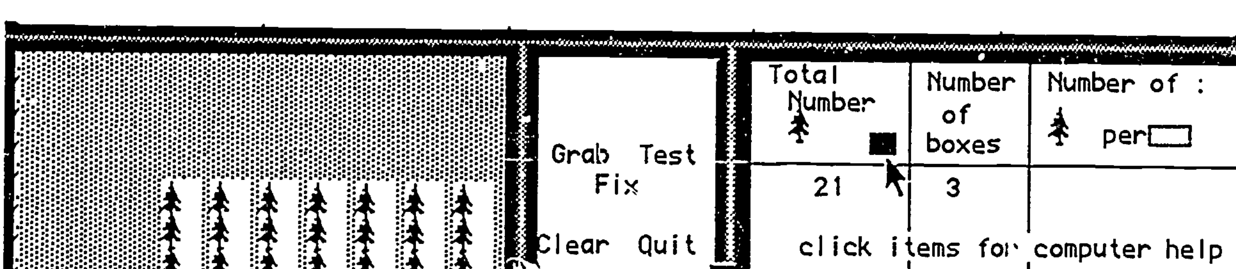

Figure 9

By additionally clicking on the Number of boxes quantity, the computer provides help by eliminating all but the correct number of boxes to be filled, as in Figure 10.

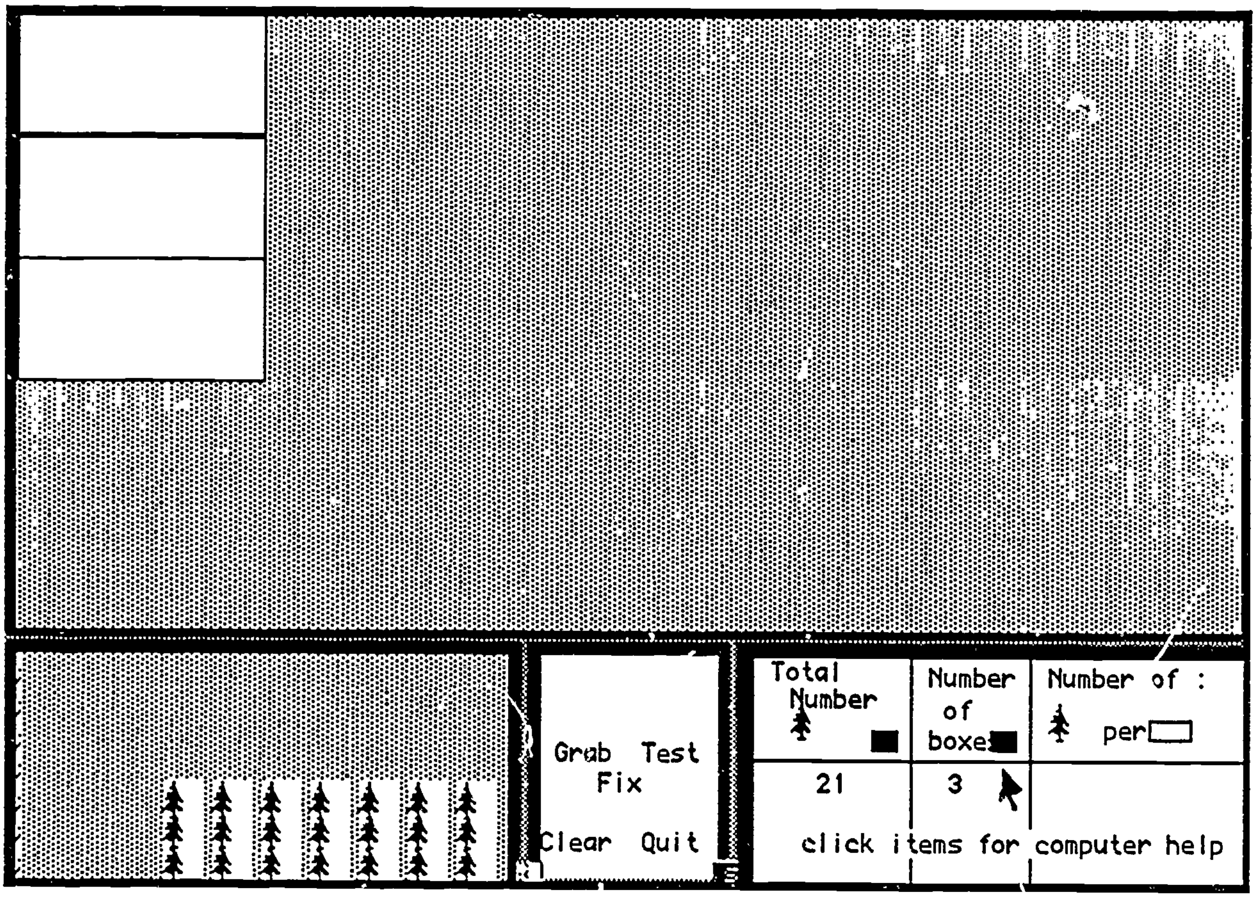

Figure 10

Then, by clicking in the command area, the user begins the process of distributing the icons in the style described in the previous subsection.

Note that the icons in the reservoir are in a 3 by 7 array, which also facilitates recognizing the underlying factor structure of the numbers. Actually, this factor structure is likely to be seen only after it has been tacitly used as a convenient organization for selecting icons to be dragged.

After each "deposit," the user can click on the Test command to get a readout in inverse video in the style of Figure 6 of the values of the quantities moved to date. (Another version provides automatic updates after each move.) Of course, the user can determine whether the problem has been successfully solved either by recognizing a correct fair share distribution in the icon window or in the data window.

If the problem had been a quotative (E'/I) division problem, then the task amounts to determining the number of boxes to be filled with the given number of icons per box from the given total number of icons. Here, if the user asks for computer help with the number of icons per box, then the scaffolding takes the form of a constraint on the "grabbing" process. The elastic rectangle that selects icons from the reservoir will "snap" to enclose only the given number per box. This supports the cognitive grouping or "unit-forming" process that we feel to be central to the conceptualization of discrete intensive quantity (Kaput, 1987). Note further that a quotative division is the first step in the "boxes strategy" for solving missing values problems as outlined in Section 1.1. This leads to the next level of the icon calculation environments, "ICE-2."

3. CONCRETELY BASED RATIO REASONING: ICE-2

Before considering elaborations, let us consider an icon-based strategy for solving missing values problems of the type considered earlier: Given that every 2 trees will shade 3 people in our park, how many people will be shaded by 14 trees?

In Section 1 we outlined the "boxes strategy" based on the provided cellular structure of the iconic representation for a discrete intensive quantity. While the "Linking Environment" used two icons in each cell, it was driven by representation-independent MORE and FEWER commands. In the environment to be introduced here, the student moves the icons themselves from within the iconic representation. The movement is an abbreviated version of the dragging used in ICE-0 and ICE-1. The fact that there are two types of icons involved is the reason for its title, "ICE-2."

The introduction to the environment is exactly like that in ICE-1, except that two icons must be chosen from the menu at the outset. Moreover, the user is again presented with the choice of using computer-generated problems, but here the problems generated are missing value problems. Let us assume that the user will input the problem. As before, clicking on a quantity label area initiates inputting a value for that quantity, and the input process is the same as before.

We will continue with our "2 trees per 3 people" intensive quantity park-planning example, so that tree and people icons have been chosen.

Here in Figure 11, providing three, or in some cases two, of the four pieces of information in the problem/data window (newly located in the upper right to make room for the second set of icons) determines the type of problem that needs to be solved. Actually, under the additional assumption that the underlying ratio of the intensive quantity is to be specified in lowest terms, two pieces of information determine the remaining two. The software defaults to this assumption if we leave the number of boxes blank. But, for example, we could model the planting of trees for picnic areas where every 4 trees shade 6 people by adjusting the number of boxes appropriately - see below.

Hence there are five different problem types:

Total Number X	Total Number Y	Number of Boxes	Y per X
?- missing values	given	implied - (but not requested)	given
given	**? - missing values**	implied - (but not requested)	given
? - multiplication	**? - multipication**	given	given
given	given	given	**?- reduced ratio**
given and/ or	given	**? - distribution/ division**	given

Two of the problem types involve determining the total number of one type of icon given the number of the other and the intensive quantity itself. These are traditional missing value problems.

A third type of problem, a dual multiplication, involves being given the ratio and the number of boxes, in order to determine the number of each type of icon, or the total number of icons taken together (which would need to be requested and tracked off line).

A fourth type involves being given the total number of each type of icon and the number of boxes in order to determine the ratio in "relative reduced form" (or "relative simplified form"). If the number of boxes is the largest common divisor of the two numbers of icons, then the underlying ratio is in true reduced (simplified) form.

A fifth type of problem, distribution or disivion, involves determining the number of boxes given the ratio and either the total number of one icon or the total number of each icon.

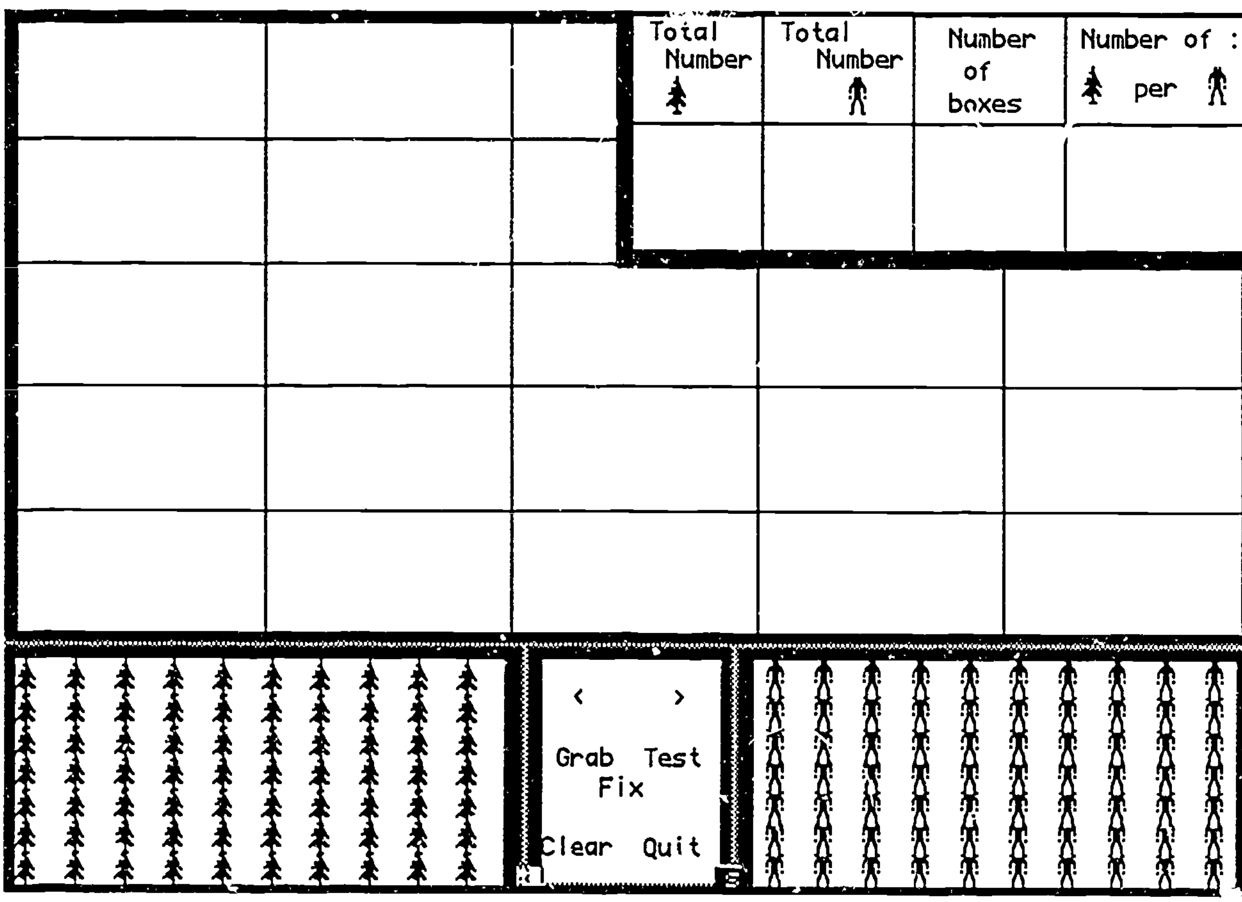

Figure 11

As in ICE-1, the user can get help relating to any of the given quantities. In Figure 12, the user has requested help for the number of boxes, the number of trees, but not the number of people.

Designating which of the two icon reservoirs to act on is done by clicking on the appropriate arrow. The arrow is pointing left in Figure 12. Selecting icons from a reservoir is done as in ICE-1. The Fix command works essentially as before: One clicks on Fix and then on the item in a cell that one wishes to move. Then one double clicks on the destination for that item, either another cell or the gray area of the appropriate (home) reservoir.

However, the means for moving icons is an abbreviated version of dragging, namely, "sending."

One merely points to the cell to which the selected icons are to be sent and double clicks.

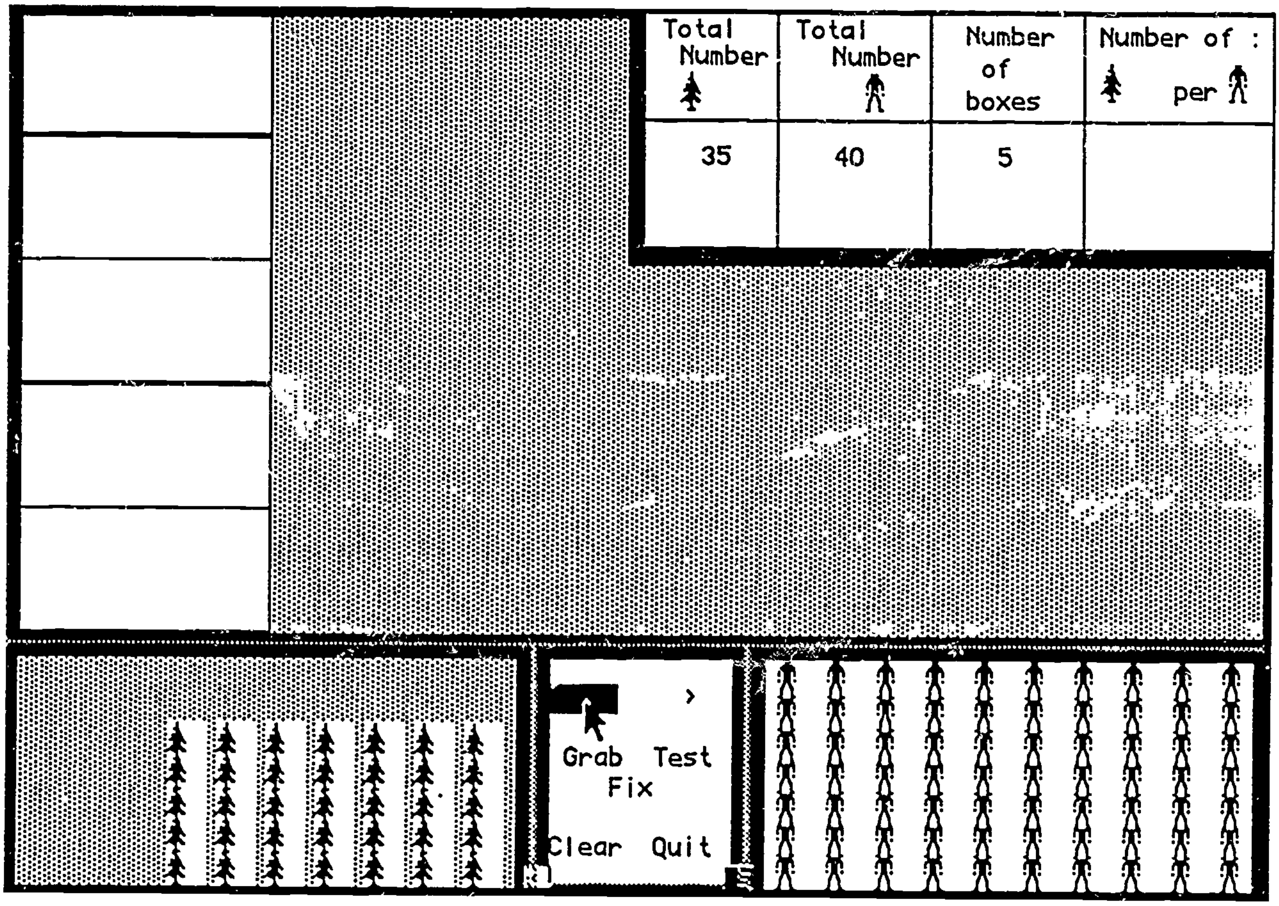

Figure 12

As shown in Figure 13, one may also choose to select icons from <u>both</u> sides before sending them. As discussed below, different cognitive processes can thus be accommodated in the solution process. Note that a running account of all quantities previously used is kept in the problem/data window. (The Test command is obviated by this convention.) Also, the "Not Equal" statement in that window is not quite appropriate, so future versions will make that statement only when icons of both types have been distributed into more than one cell.

More than one distribution strategy is possible in solving a traditional missing values problem, depending in part on the conditions under which the problem is solved. For example, the "boxes strategy" has a perfectly concrete embodiment in this context:

(1) The division step: distribute 14 trees into cells, grouped 2 per cell, yielding the layout in Figure 14,

(2) The multiplication step: match each of the groups of 2 trees with a group of 3 people,

(3) The counting step: count the number of people icons, using either the gray shadows in the people icon reservoir or the people icons in the cells.

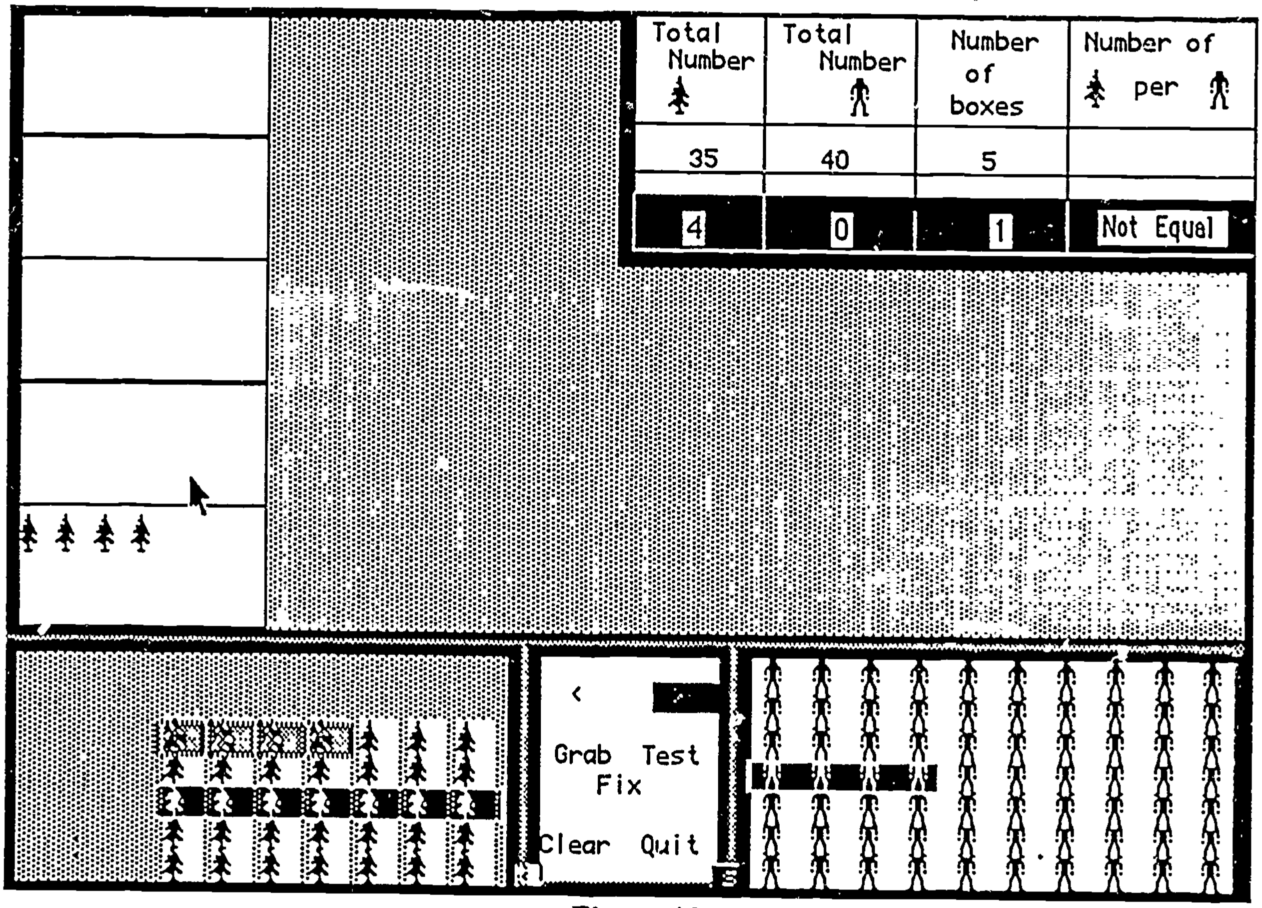

Figure 13

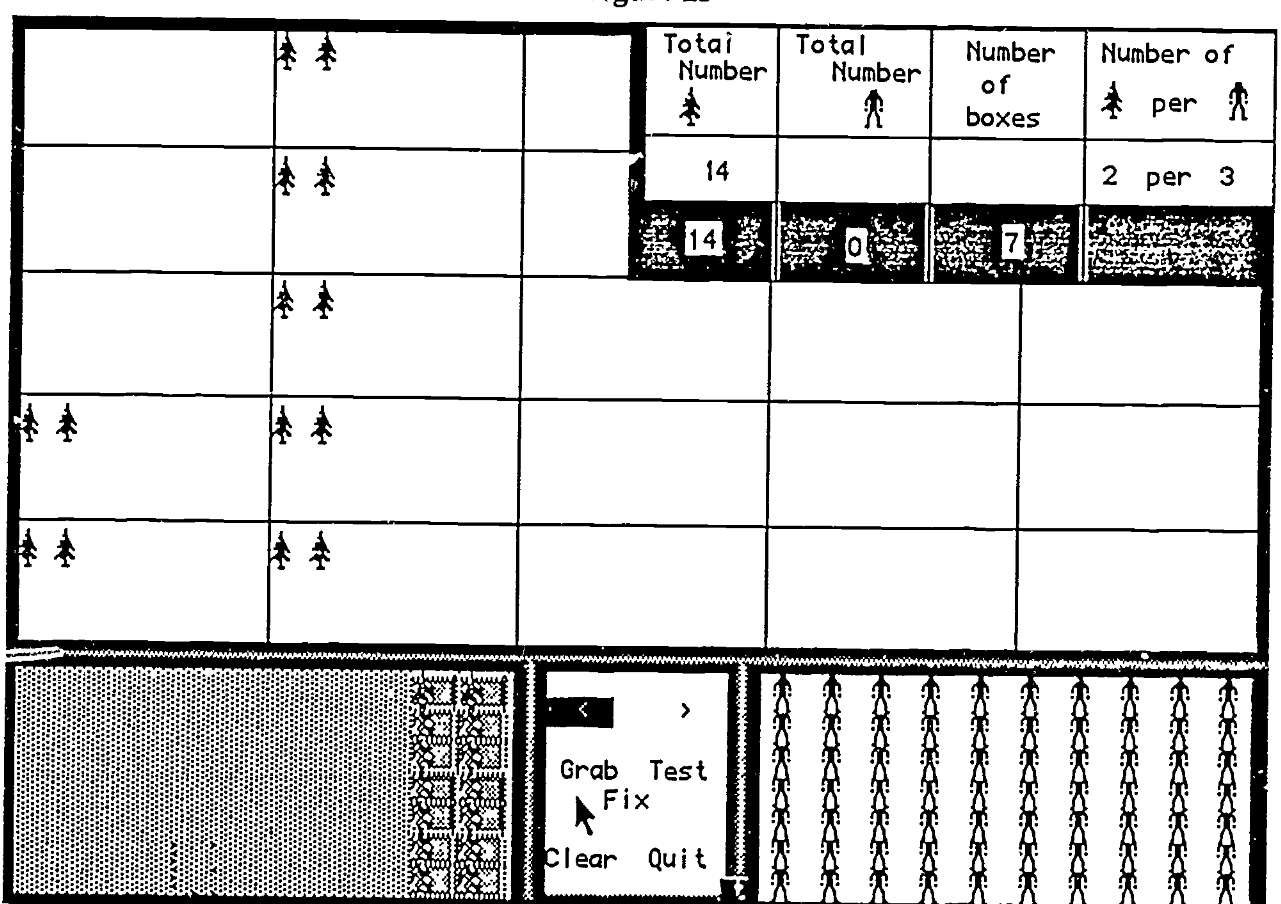

Figure 14

The layout in Figure 15 is a depiction of step 2 *in medias res*. Note that the user had chosen the "computer help" option for the number of trees.

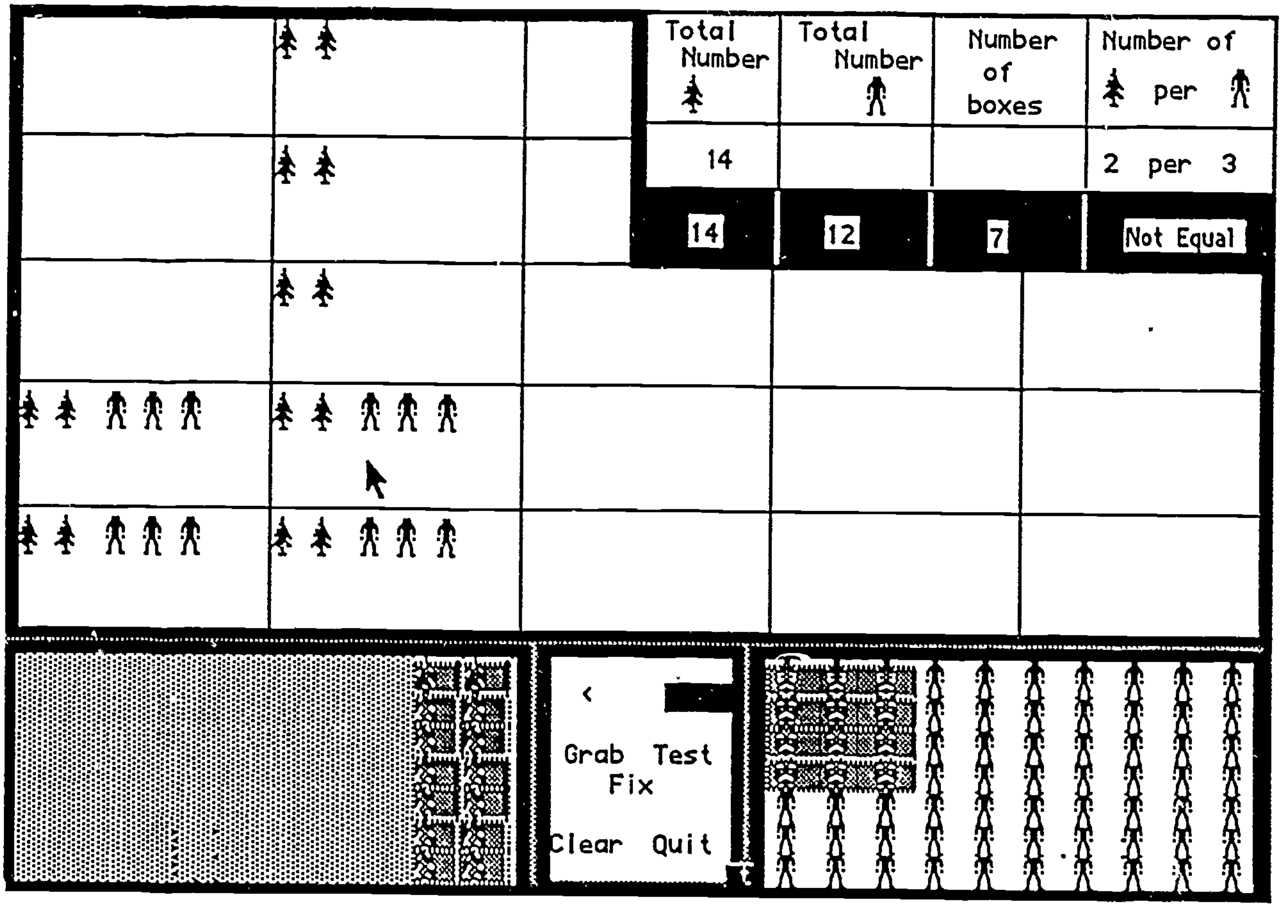

Figure 15

The different concrete primitive strategies that are possible for solving missing values problems amount to variations on the assembling of the three components: grouping, matching, and counting. Moreover, the grouping component can be scaffolded using the "computer help" option. This can be accomplished by clicking on the ratio box - the rightmost column in the problem/data window. When this option is chosen, the grouping process is constrained so that only the "correct" number of icons can be selected from either reservoir in a single "grab."

4. LINKING THE ICONIC, NUMERICAL AND GRAPHICAL REPRESENTATIONS

Linking/Incrementing Environments

- Iconic Representation
- Numeric Representation (table of data)
- Graphic Representation (coordinate graph)
- Algebraic Representation (linear equations)

Chart 2

4.1 Introducing the Table of Data Using The Linking Environment.

Figure 3 in Section 1 depicts the Linking Environment with all four representations active. In our use, the new representations have been introduced one at a time, beginning with the numerical - the table of data. This is to control the degree of novelty that the students encounter. We have done this by asking students to record on paper the number of icons of each type that are enclosed in systematically increasing groups of cells - where the sequence of icon cells are taken from screen dumps of the type depicted in Figure 2 of Section 1. They are then presented with the Linking Environment with only the icon and table of data windows visible (with the same intensive quantity, of course) and asked to compare their table with that generated by the computer as they click on the MORE button. A typical screen can be seen in Figure 16

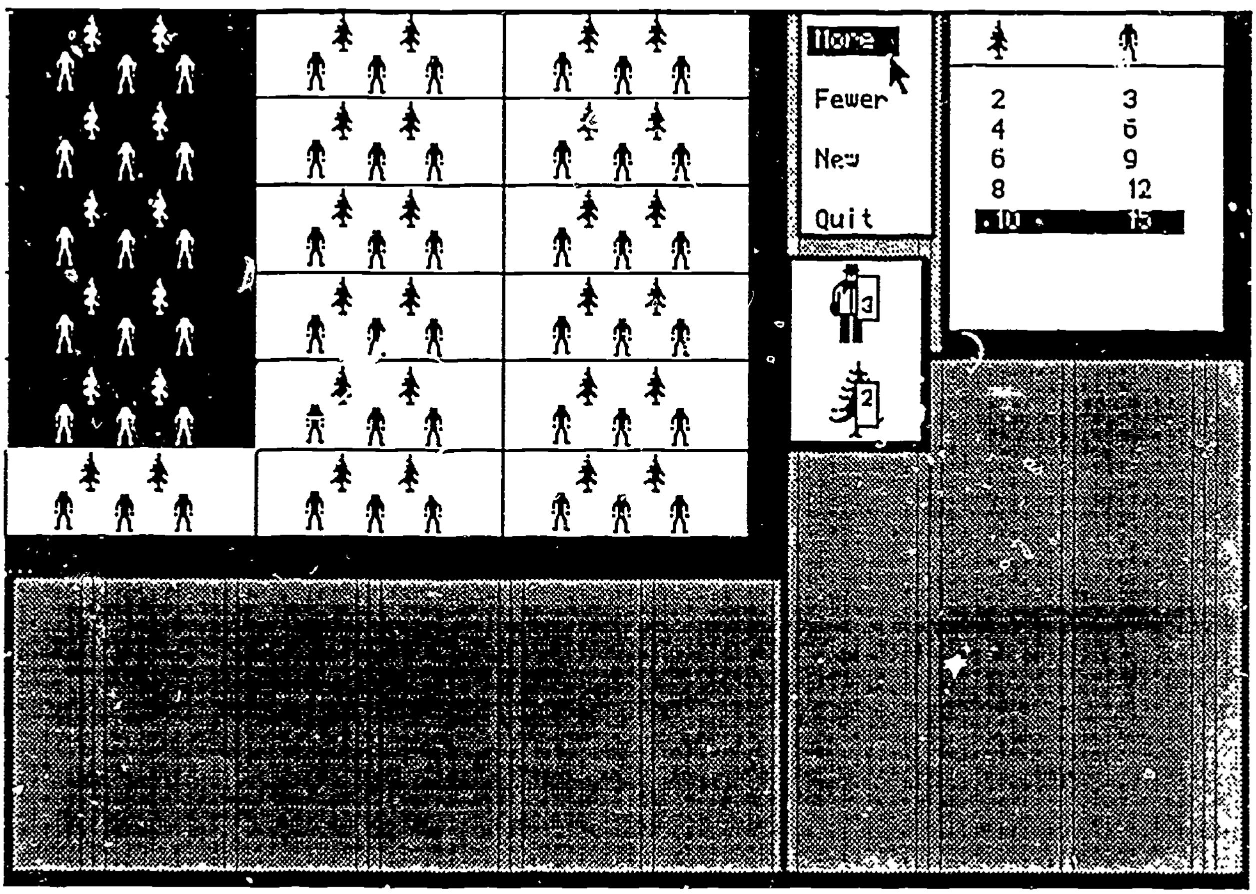

Figure 16

The table of data scrolls when its window is filled. By clicking on FEWER, the highlighting process works in reverse, but the number pairs remain in the table.

4.2 Traditional Introductions to the Coordinate Graph.

Several approaches to introducing the coordinate graph are possible, depending on the experience of the students. If the students have some experience with coordinate graphs, then an introduction similar to that for the table of data can be employed. In that approach, students plot points on a scaled paper coordinate graph based on a given table of data and perhaps the same kind of sets of icon cells as used to introduce the table of data earlier. They are then given the Linking Environment with the table of data and coordinate graph actively linked to the iconic representation and asked to compare their graph with that generated by the computer.

Among the versions available, one scrolls the coordinate graph in the same style as the table of data

scrolling - the viewing window slides up along the line of points so as to include the point with the largest coordinates. The boxed point in the coordinate graph always corresponds to the highlighted pair in the table. See Figure 17. (The version shown does not include the algebraic equations.)

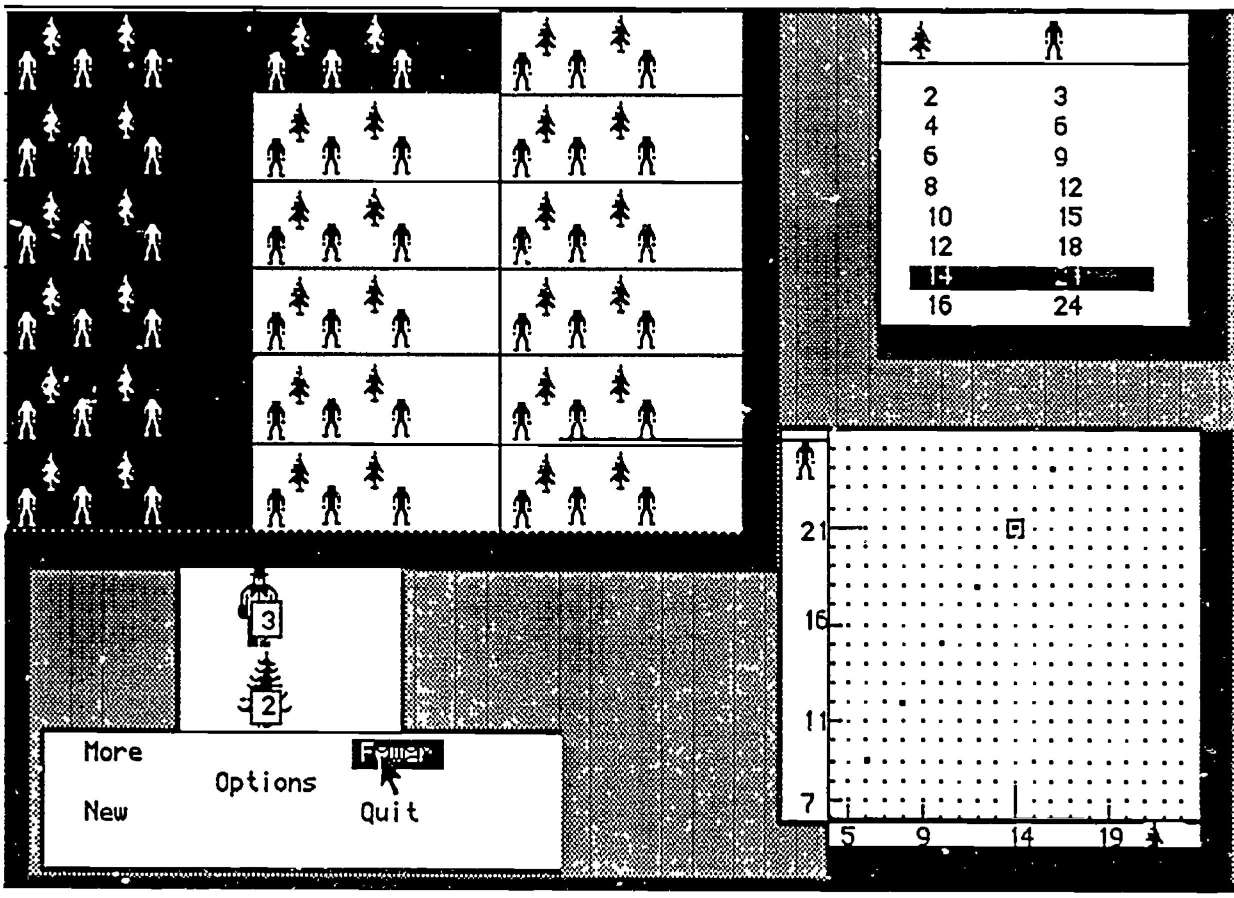

Figure 17

4.3 Transitions to the Coordinate Graph Representation.

For students with little or no experience with coordinate graphs, we designed a simple coordinate graph which makes more explicit the relation between the iconic representation and the coordinate graph. As the student clicks the MORE button, icons are <u>deposited</u> on the respective axes in groups corresponding to the sets highlighted in the icon window, i.e., according to the given ratio. This is shown in Figure 18. Of course, this graph cannot scroll. Instead, if one continues the incrementing process, the effect is that the points "arch up and off the screen." Not implemented as of this writing is a version that also labels the coordinates of the highlighted (boxed) point and the axes.

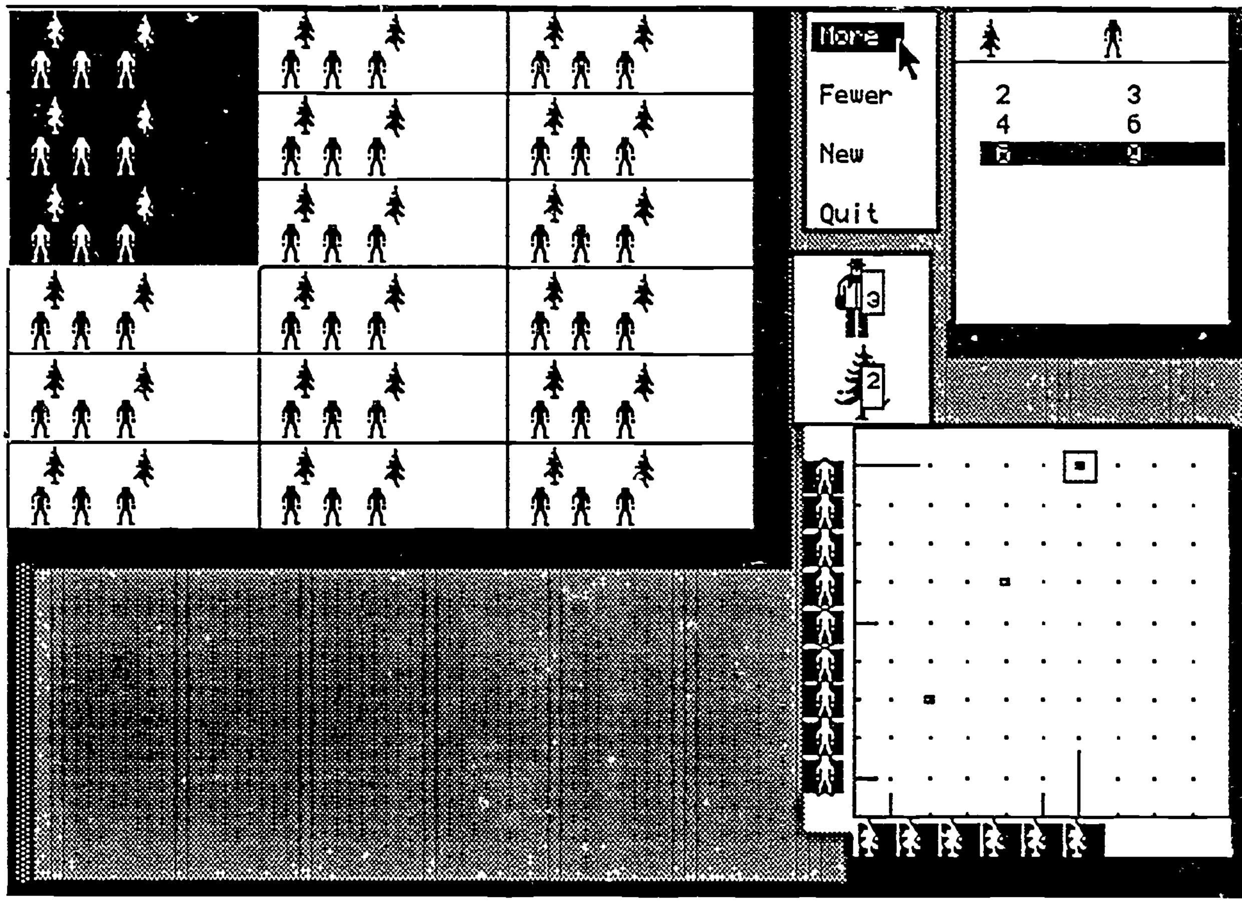

Figure 18

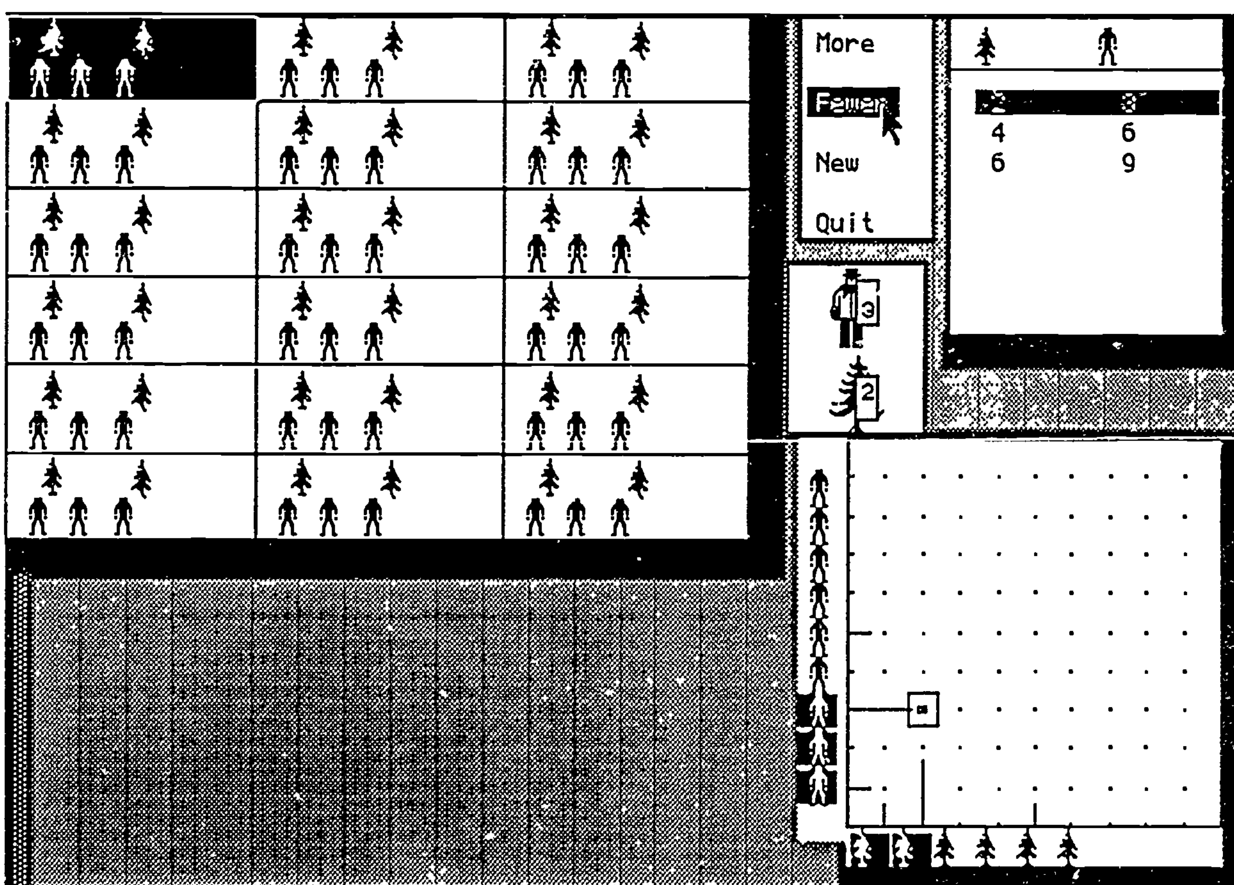

Figure 19

In Figure 19 we show the results of clicking on the FEWER button: The "dehighlighting" process for the icons in the cellular representation and the table of data is paralleled in the transition coordinate graph - but the previous points are deleted in this case.

In Figure 20 below, we illustrate the Linking Environment with the iconic representation toggled off and all the equations active. Note that the lower left equation is the standard slope-intercept, or "y = mx + b" form of the equation of a straight line.

The algebraic representation is the newest of our environments, hence is the least tested and least refined. Subsequent versions will have equations that are easier to read, and will likely be able to be turned off individually in the same style as the other representations, via a toggle on the boundary of the subwindow. Depending on a teacher's goals, this will enable one to emphasize a single form of the equation as well as illustrate the equivalence of all the equations by displaying the common values of the quantities across all the equations and the expressions in those equations.

This Linking Environment has been elaborated to support separate, independent incrementing of variable values. Thus, MORE has been replaced by MORE "X," MORE "Y," MORE BOTH. This provides a whole new range of flexibility. For example, by incrementing the variables alternatively and plotting the corresponding points after each increment, one is able to see explicitly the "rise-run" process associated with changes in the respective variables. By incrementing only <u>one</u> of the variables, one can see the resulting horizontal or vertical path of the points.

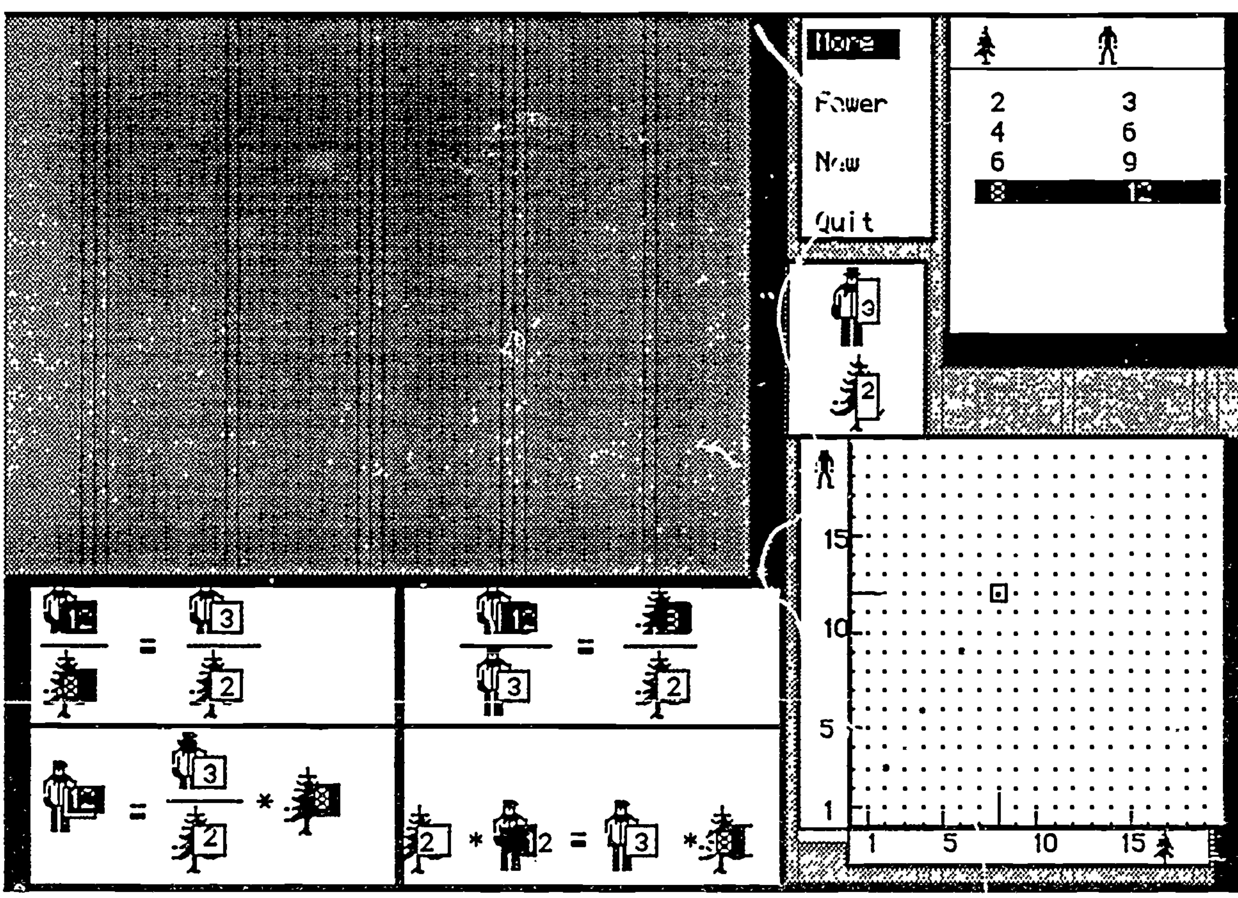

Figure 20

5. MISSING VALUES PROBLEMS ACROSS REPRESENTATIONS

Missing Values Environments

- Icon Input
- Numeric Input
- Graphic Input
- Algebraic Input

Chart 3

In our next environment, designed specifically for posing and solving missing values problems, the overall appearance and initial actions taken by the student to specify an intensive quantity are similar to that of the Linking Environment. But instead of driving it from representation-independent MORE and FEWER buttons, the user can drive it via inputs to <u>any</u> of the four representations. The user can simultaneously view the consequences of such inputs in any of the other representations. We illustrate with examples, all based on our familiar 2 trees per 3 people intensive quantity.

As in the previous problem solving environments, the user is offered a choice of whether the computer or the user should generate the problem. However, here the choice is simply whether the computer provides a value of one variable and the user matches it, or whether the user inputs values of both variables. Hence if the computer generates problems, there are only two types of problems: the computer provides the number of trees and the user determines the number of people, or vice-versa. The numbers provided are multiples of one or the other number chosen at random of a size that will allow an iconic representation. The user always chooses the representation in which the input takes place.

When the user generates the problems (usually from a story provided in written materials or a classroom discussion), the user is responsible for inputting the given number. That inputted number then appears in any of the representations that are active at that time, as does the second number, the "answer number." Of course, the representational form of the inputs is not necessarliy an alphanumeric character, but varies by representational type.

We will now assume that the user inputs problems, and we shall characterize the inputs via the representation in which they are initiated. The choice of order of representation in which these inputs are made will likely vary according to the interests and needs of the class.

5.1 The Table of Data.

Here the user merely types in the numbers at the bottom of the table of data as indicated in Figure 21. In this Figure as in those following we have added an overlay to emphasize which window the inputs occurred in. Here we also circled the second number, the "answer number."

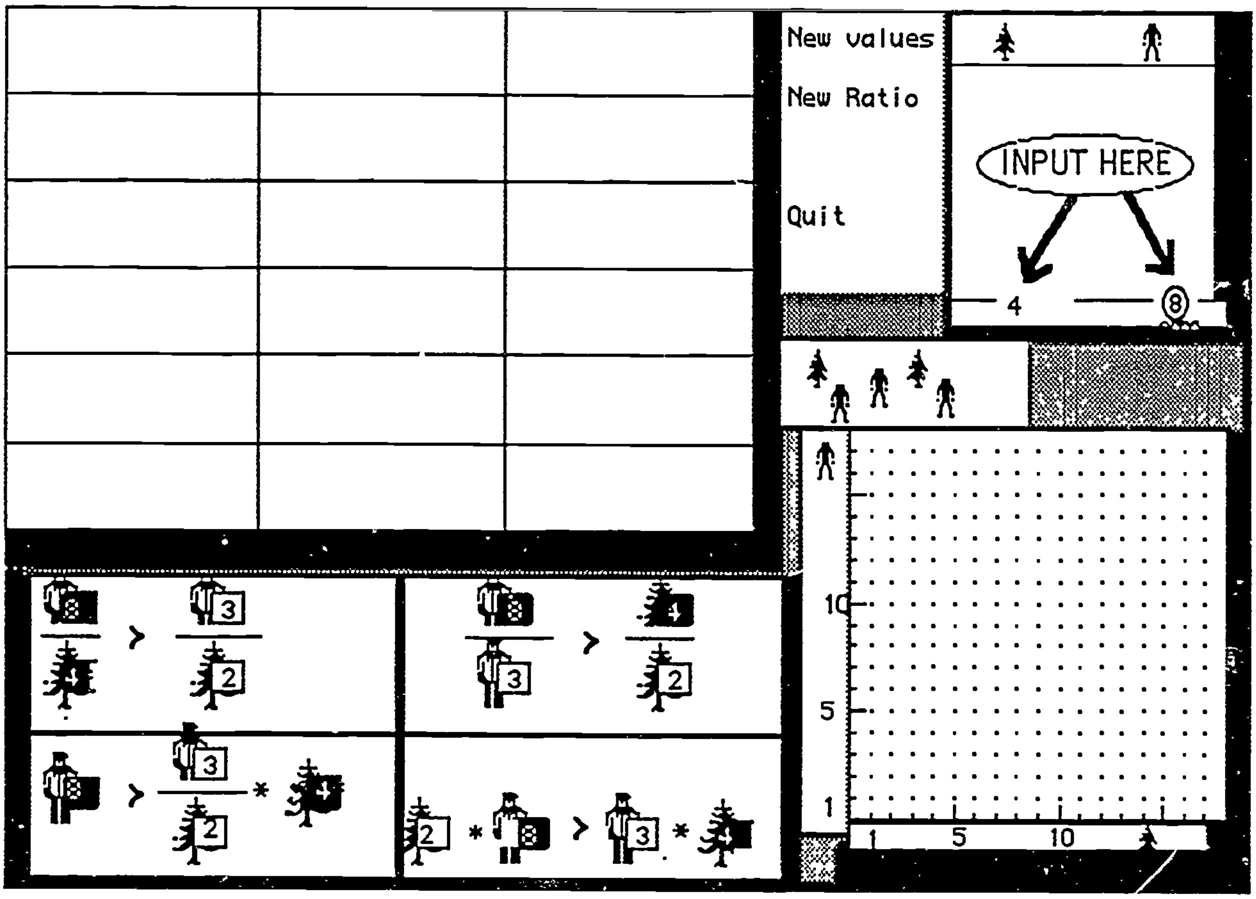

Figure 21

Notice that the inputted values appear in each equation of the algebra window, which is active. Note further that the second number, 8, did not yield equalities in the algebra window, but inequalities. The other two windows are not active. If the icon window were active, the computer would fill in as many cells as possible identical to the model cell, so an inappropriate input results in cells that do not match the model cell, as in Figure 24 below (where the inputs were 6 and 8). Of course, the computer merely reports the student's input - the student evaluates this report.

5.2 Algebraic Equations.

Here, one inputs in one of the algebra subwindows, designating which one by pointing and clicking on the appropriate variable in the appropriate subwindow - - in Figure 22 (the upper left subwindow in this case). The system responds with a question mark, which the user types over to input the first number. Before any numbers are input, in place of the comparison operator (the equality or inequality symbol) is a gray rectangle. After the first number is input a question mark appears in the second variable of that expression to indicate that an input is expected there.

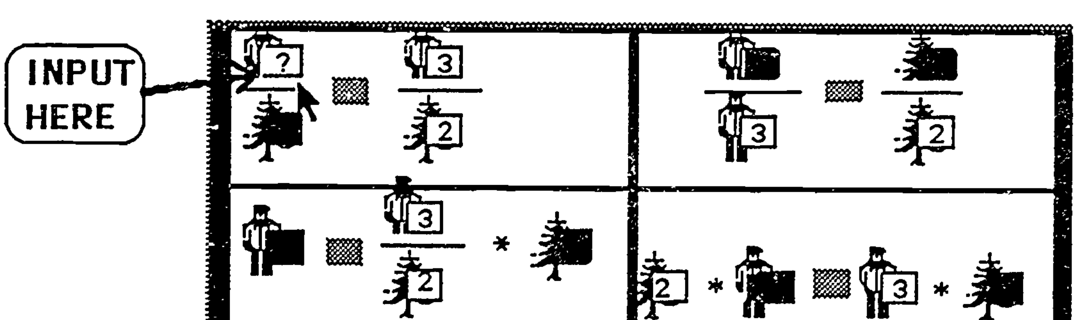

Figure 22

When the second number is input (which must be done in that same subwindow), the appropriate comparison symbol appears in all the active subwindows, as in Figure 21, and each of the other active windows displays the two numbers as well.

5.3 The Icon Window.

In earlier implementations of the Missing Values environment the icon window was a primary source of feedback regarding inputs made in the more abstract representations. Now it also receives input from the user, so it is now read-write in nature. Here the input form is an abbreviation of that in ICE-2. One simply identifies the icon to be deposited by clicking on that type of icon in the command window. One then clicks in those cells where the computer is to deposit icons of that type. The computer drops a set of that type of icon matching the set in the original model. However, a given cell will accept only that set. Surplus clicks in a cell yield no response. One can change the type of icon being deposited by clicking on the other icon in the command window. The option to deposit both icon sets simultaneously is also provided. Any number of cells can be filled, or partially filled. In particular, the user (which always potentially includes a teacher in a teacher-centered classroom situation) could deposit icons of only a single type in order to match a problem situation, e.g., "How many people will be shaded by 14 trees in our park?". All active windows are continuously updated after each click in a cell.

The icon representation is not quite parallel to the other representations in the degree of input freedom offered the user because the values of the variables being implemented are limited to multiples of the values comprising the given intensive quantity. Beyond this, the major constraint is on how those values are put into the representation (to match the initial pattern that was used to input the intensive quantity at the outset), just as the style of input is constrained in any of the other representations. The difference is that we are allowing the user to increment both variables, perhaps alternatively, rather than incrementing a single variable at a time. This new flexibility is important because it facilitates different strategies for solving missing values problems. Figure 23 depicts an incorrect (or at least, incomplete) iconic input viewed in all three of the other representations.

A decision on how to deposit icons was based on the assumption that the students using this environment had already had experience in the ICE software. If a group of students were to be introduced to missing value problems only in this multiple representation environment, then we would likely want to have a free-deposit mode, where the student could drop icons in cells at will, in any order. This might occur if the software were used with a group of 6th-9th graders, for example.

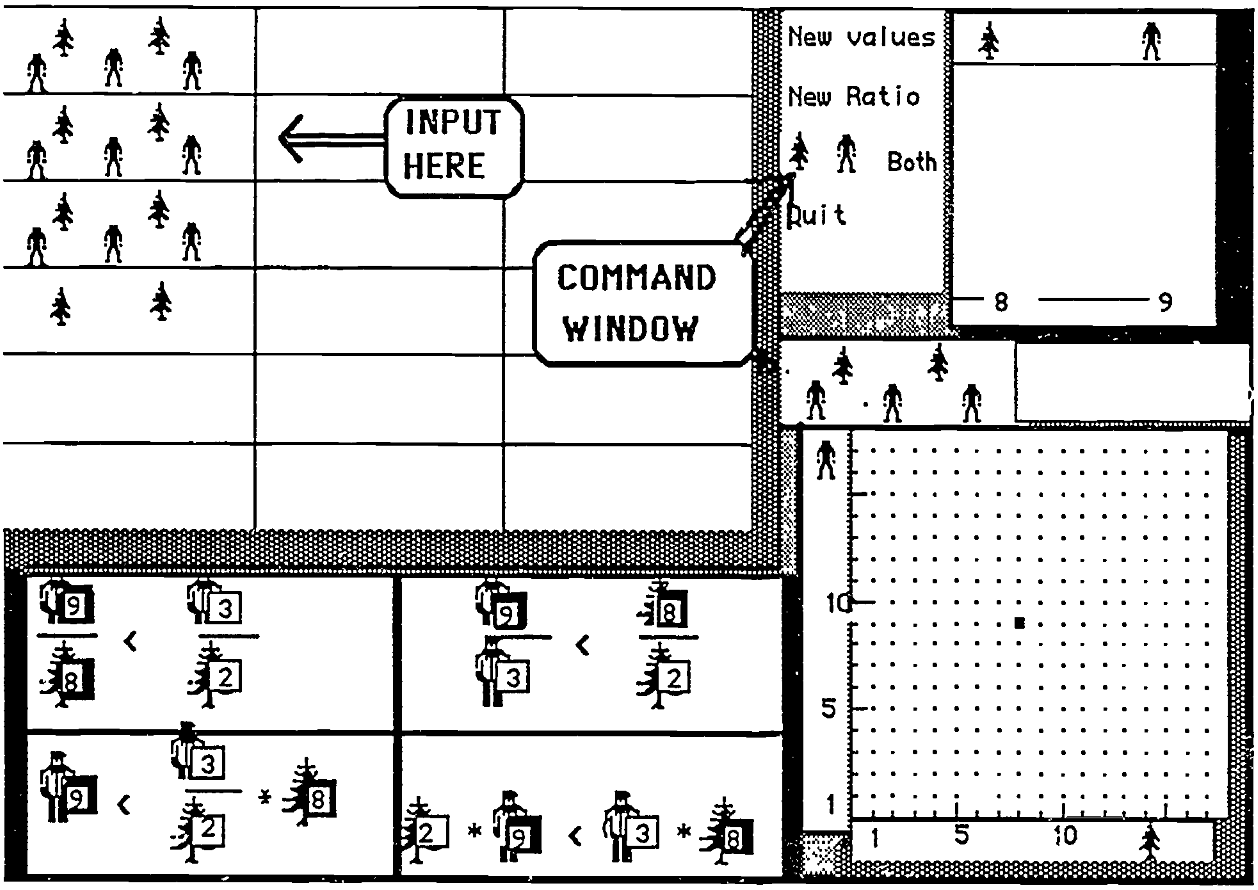

Figure 23

Note that incrementing is the only style of input that is natural in the iconic representation. While incrementing could be achieved in the other representations, e.g., using ICE-style up-down arrows in the table and the algebra windows, we have decided to use the input style that is conventional to a particular representation. Recall that the Linking Environment uses incrementing across all representations via the MORE and FEWER commands.

5.4 The Coordinate Graph.

Here the value of each variable is specified by an appropriate horizontal or vertical line: in our example, the number of trees is specified by a vertical line and the number of people by a horizontal line. One inputs values by sliding the intersection of the vertical and horizontal lines using the mouse. A click serves to fix the values (position at the intersection) of both variables as in Figure 24, where the algebra window is toggled off. If the computer has generated a value, then that value is represented by a fixed vertical or horizontal line. In that case the user has control over the other line to fix the value of the second variable.

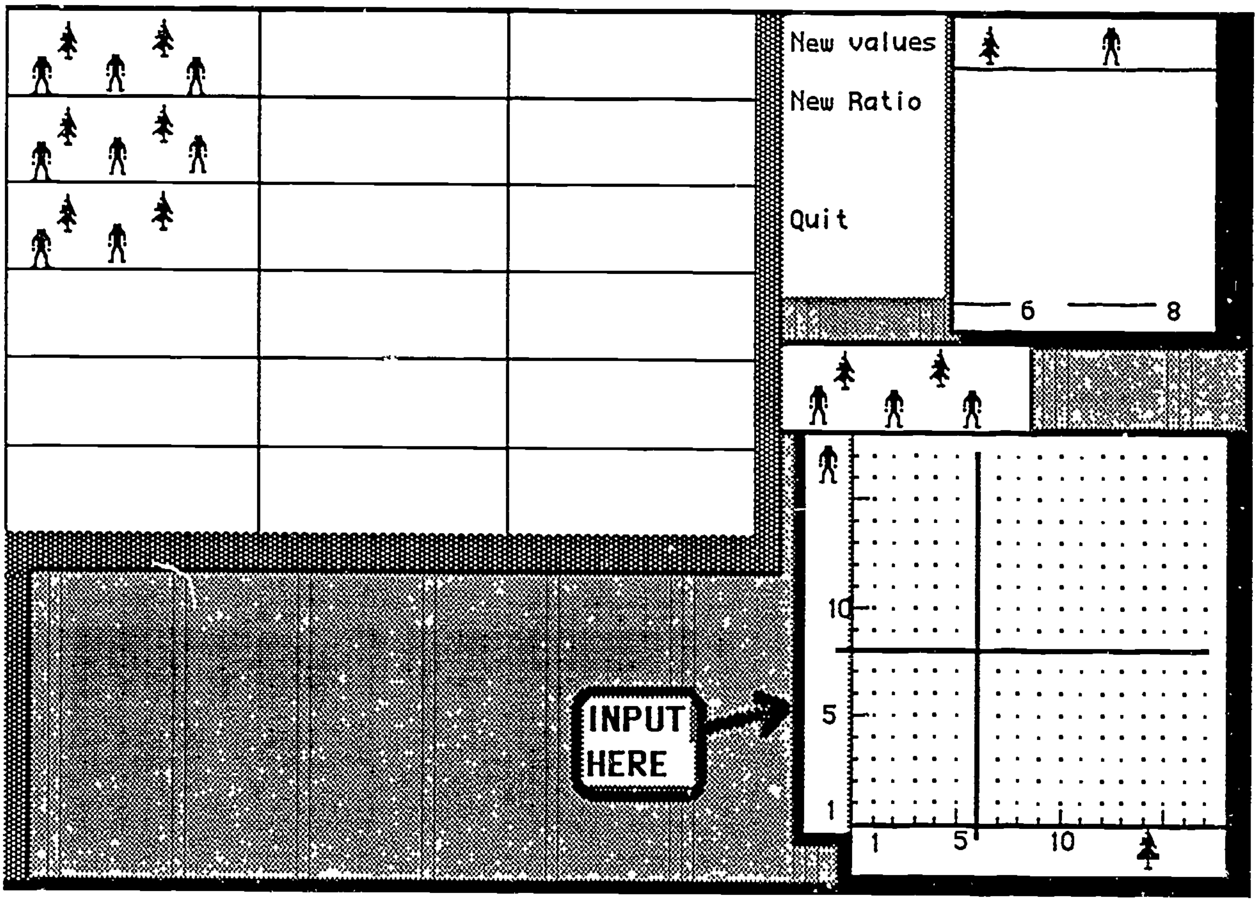

Figure 24

5.5 Recording Results of Previous Problems.

As can be seen in most of the previous figures, there are two options after completing a given missing values problem: either do another problem with the same intensive quantity and change the values (by clicking on the New Values command), or change the intensive quantity (the New Ratio command). If the latter were chosen, then the original icon-menu screen appears and one begins anew by picking icons, etc. If the former "new values" option is chosen, then one of two things will occur: (1) if the solution was incorrect, then the number pair will disappear from all representations when new values are entered in any window, or (2) if the solution of the previous problem was correct, then the number pair resulting from the solution of the immediately previous problem will be stored in those representations admitting the simultaneous representation of more than one value of a variable, namely the table of data and the coordinate graph. In the table, the number pair is stored in the body of the table, as opposed to inputs, which appear initially at the bottom. They get "promoted" only if correct. Points are stored in the coordinate graph as bold points, whereas "tentative" values resulting from inputs to that representation appear as less bold points, until they are promoted. This resolves the issue of what to do with incorrect pairs.

The differences in a single representation's capacity to store and simultaneously display multiple values of variables turns out to be important here - neither the iconic representation nor the algebraic comparison statements, including equations, have this capacity. Repeated copies of these representations would be required to do this.

Of course, after several problems have been correctly done, then accumulated feedback in the tabular and graphical representations becomes informative, especially in the coordinate graph because of the linearity of the set of points resulting. Another option available for the table of data

provides automatic ordering, which renders the pattern in the numerical data more salient. (Note that a coordinate graph is "self-ordering.")

5.6 The Locus of the Reasoning/Computing Processes.

There is a subtle shift as one moves from the icon calculation environments to the missing value environments. On one hand, the procedures executed using icons are very visibly and explicitly instantiated. On the other hand if one is solving a missing value problem using numbers, as in the table of data, then one's computations are done off-line -- either in the pure cognitive medium or with extra-cortical aids, such as pencil and paper or calculator. One then reports the results of those computations to the computer. This design is deliberate. The iconic representation is the place where the cognitive structures guiding the computations are generated, as internalizations of those actions on icons.

5.7 Introducing the Table of Data Using Missing Values Problems.

Another environment has been developed that supports the systematic solution of missing value problems using either numerical or iconic representations. After picking icons from the icon menu, one is presented with a screen such as that in Figure 25 which can be used to build a "model cell" in a style and with a result similar to that in ICE-2: select and then send icons from the reservoirs to the empty cell below the blank table of data. (Note that, as in ICE-2, the user has no choice in the layout, only the numerosity of the icons, in the model cell.) This specifies the intensive quantity for action.

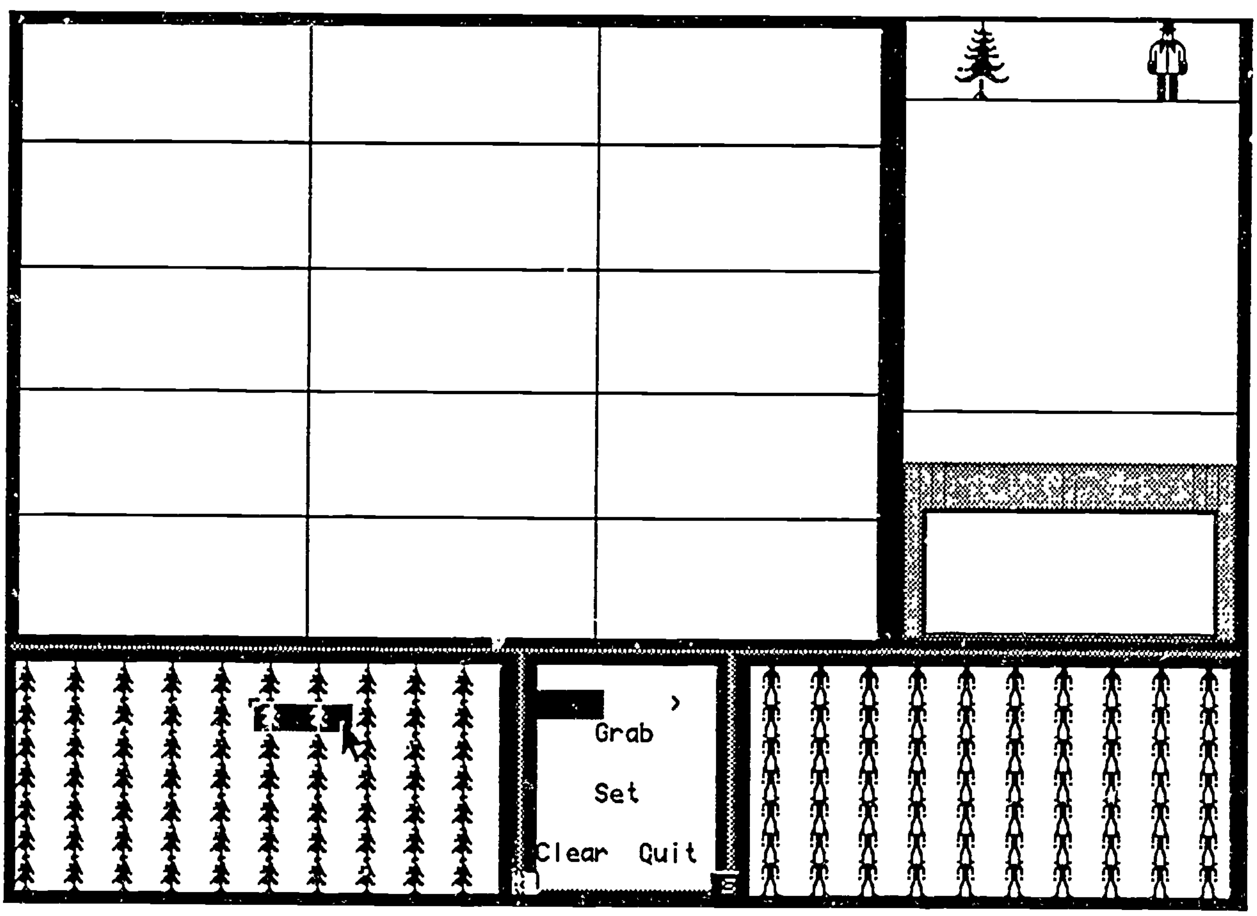

Figure 25

When the intensive quantity has been specified (which is signalled by clicking on "Set" in the menu

box), then the user can set up and solve missing values problems using either the table of data (by first clicking on Table in the menu box in Figure 26) or the icon window.

If the icons are used, the student merely double clicks on the cell in which icons are to be dropped and the computer deposits the first set of icons - in our case, 2 tree icons. A second double click in that cell drops the other set of icons - 3 people icons. If the second double click were made in a new, empty cell, then 2 tree icons would be deposited in that new cell. Surplus clicks in a cell yield no response. This is the default order for depositing icons. One can specify the icon to be dropped by first clicking on the appropriate type of icon in the model cell. However, the icons will be deposited in the same position as the model cell. The decision to deposit groups of icons rather than singletons was intended to the support of the cognitive grouping operation - grouping the 2 trees and the 3 people into conceptual entities that can then be joined into the conceptual entity which is the intensive quantity itself.

Whenever an icon appears in a cell, a corresponding icon is "grayed out" in the reservoir and the entry at the bottom of the table is updated. A new version of this environment will accept inputs from wherever the user indicates, simply by pointing and clicking. This will eliminate the need for the Table command.

If one were to use the table, i.e., use numerical inputs, the computer deposits icons after every entry. In Figure 26 we show the result of entering 20 in the tree column.

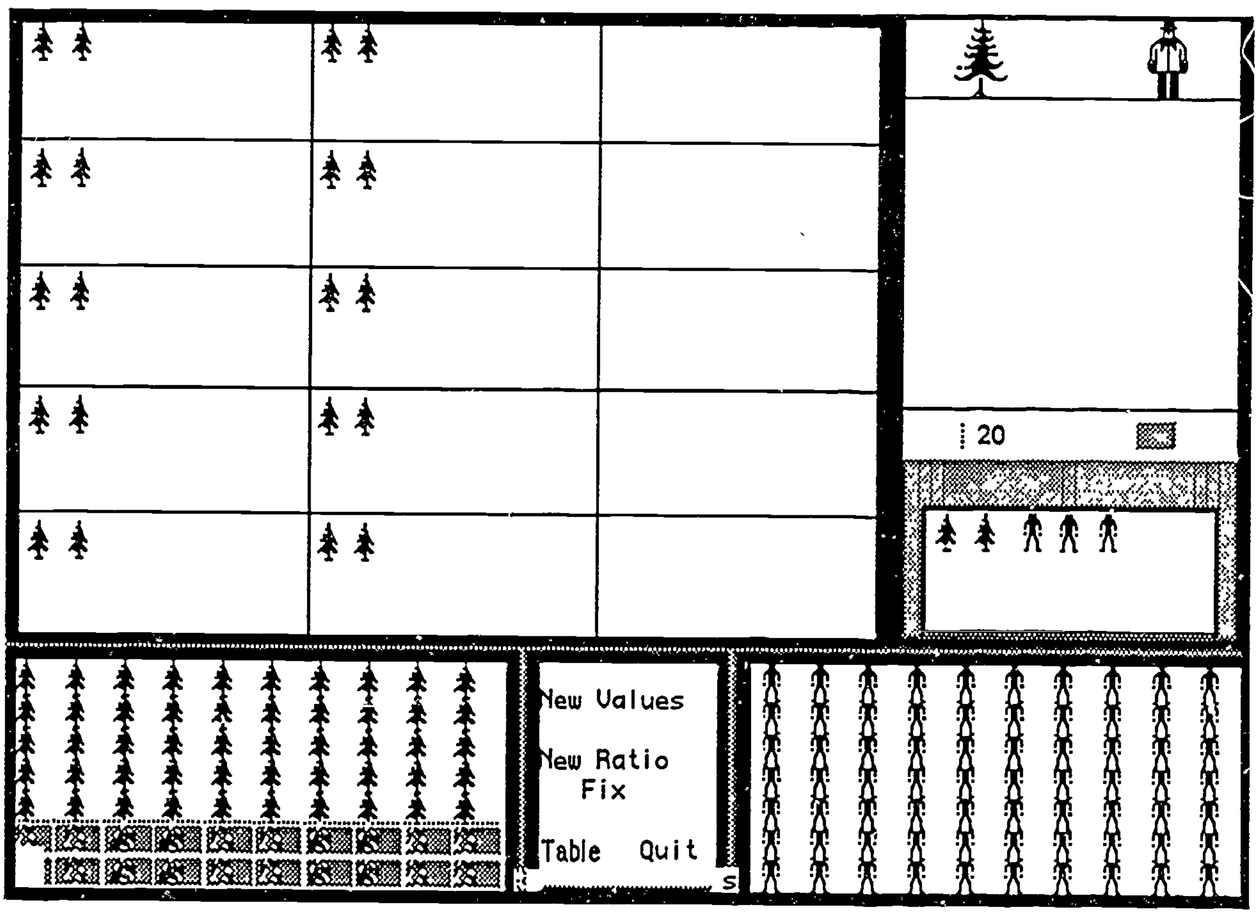

Figure 26

After entering a number in the table, the user can finish the problem (How many people will be shaded by 20 trees?) using the icon window. A double click in each cell with tree icons will deposit 3 people icons, as indicated in Figure 27, where three cells have been filled using this

procedure. (There is also available, here as elsewhere in our discrete environments, an auditory

signal - a beep - for each increment of a variable value.) Note the update in the table. This environment provides a very efficient consolidation of the actions taken in the ICE environments. It also provides a transition to the multiple representation missing values problem environment that has been described in the first several parts of this section.

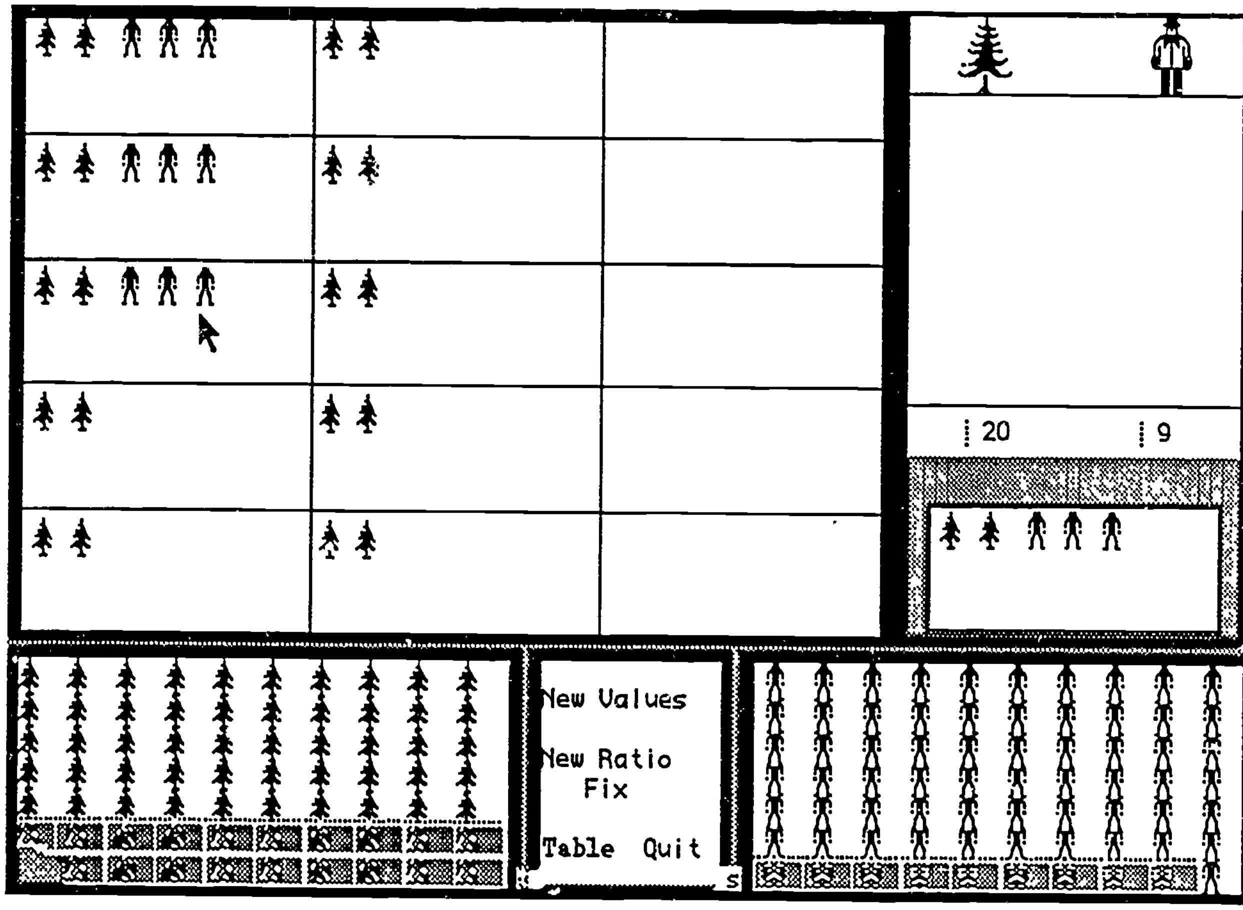

Figure 27

The purpose of the iconic representation in that missing value environment is as a last accessible link between those concrete actions and the increasingly abstract coordinate graphical and algebraic representations.

6. SAMPLING ENVIRONMENTS.

Sampling and Comparison Environments

- Sampling
  - Regular/Homogeneous Samples
  - Irregular Samples
- Sampling and Comparing Two Intensive Quantities
  - Regular
  - Irregular

Chart 5

6.1 The Two Other Aspects of Intensive Quantity: Homogeneity and Order.

Each of the previously described environments deals with what we have called the multiplicative structure aspect of intensive quantity. This aspect involves the use of intensive quantities in multiplication and division operations. Another aspect of intensive quantity is the basis of its use to model intensive attributes of situations, entities, or substance, e.g. speed on a journey, number of cars per household, density of grams per cubic centimeter. In particular, to describe such an intensive attribute using a constant intensive quantity means that the attribute is possessed homogeneously - it is constant across samples, as opposed to, say, a nut cake, whose density varies from point to point. It is this homogeneous attribute modeling aspect of intensive quantity that our next set of environments engages.

In the discrete world, homogeneity means homogeneity down to a minimum sample size below which the sampling process is inappropriate. Hence, for example, in the trees-people park example that we have been using, we know that every picnic area will have 2 trees for every 3 people that it is to accommodate. This does not mean that every picnic area will have only 2 trees and 3 people associated with it, a view that must be explicitly discussed with students as part of the context setting portion of the problem. Unfortunately, this is the view encouraged by the cellular iconic model that we have been describing. That particular iconic model has features which, while they support a very useful set of reasoning patterns, lead one astray by providing "boundaries" that do not necessarily correspond to boundaries in the situation being modeled. Indeed, one lesson from attempting to build useful representations of a complex idea such as intensive quantity is that different representations have different strengths and weaknesses, and that more than one representation is necessary to reveal all aspects of such a complex idea. We should mention that these considerations led to the deletion of boundaries appearing in the highlighted sets of icons in the linking environments. The highlighted icons appear as a solid block. (See Figures 16-18.)

Thus we have built a second type of iconic representation that addresses this issue and provides the basis for dealing with this second aspect of intensive quantity. (For more on the complexities of this modeling issue, see (Kaput, 1987).

One type of environment involves sampling and recording the results in tables of data. The other involves comparing the relative size of two intensive quantities having the same units using both tables and coordinate graphs. This latter environment engages the third aspect of intensive quantity, order. Intensive quantities having the same units can be compared. For example, one might ask of several tree-people intensive quantities describing different parks, which is shadiest? That is, which has the most number of trees per person? The representation for efficiently comparing intensive quantities is not the iconic, but rather the coordinate graph, where the magnitude of the intensive quantity is represented by slope, a visually salient and unitary feature. The table of data can be useful if one can find pairs that share values in a particular column, leading to the unit rate comparison idea.

6.2 Simple Sampling Environments.

The environments described so far assume that the student is provided the underlying ratio (often, but not necessarily in reduced form). A different icon-based environment has been constructed that uses a window tesselated with the two kinds of icons in such a way that by stretching an elastic rectangle to enclose portions of that window one is guaranteed of enclosing icons in a constant ratio. The student can slide such a "sampler" around the window to effect other samples, or change the size or shape of the sampler to obtain yet other samples. The result of each sampling act is stored in a table, which the student can use to infer the underlying ratio as in Figure 29. Note that the icons are not displayed in rectangular cells - the homogeneity is achieved differently.

The screen shown in Figure 29 follows the choice of ratio made in Figure 28, which is what appears after the user makes choices from the icon-menu screen.

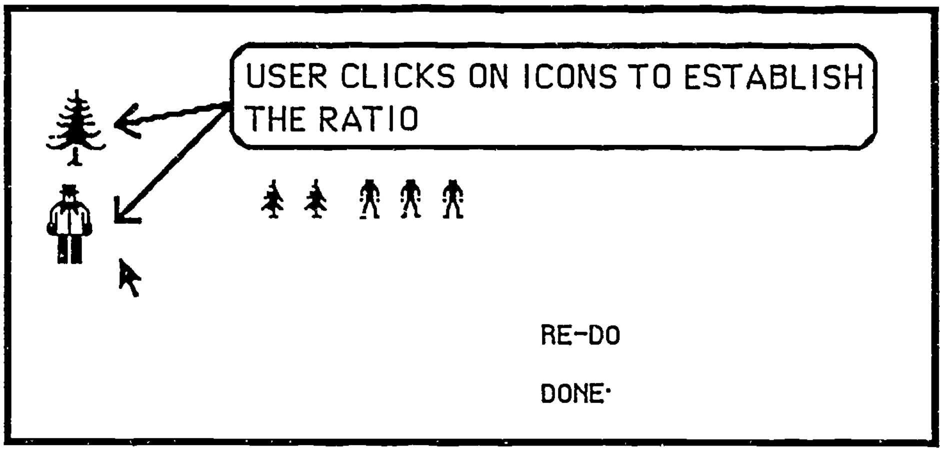

Figure 28

In this particular option, the user knows the ratio and the sampling process is entirely regular in the sense that the stretched elastic rectangles will snap around only representative samples, i.e., they all match the given ratio.

In Figure 29 we have stretched a single rectangle and clicked, which produced a single entry in the table.

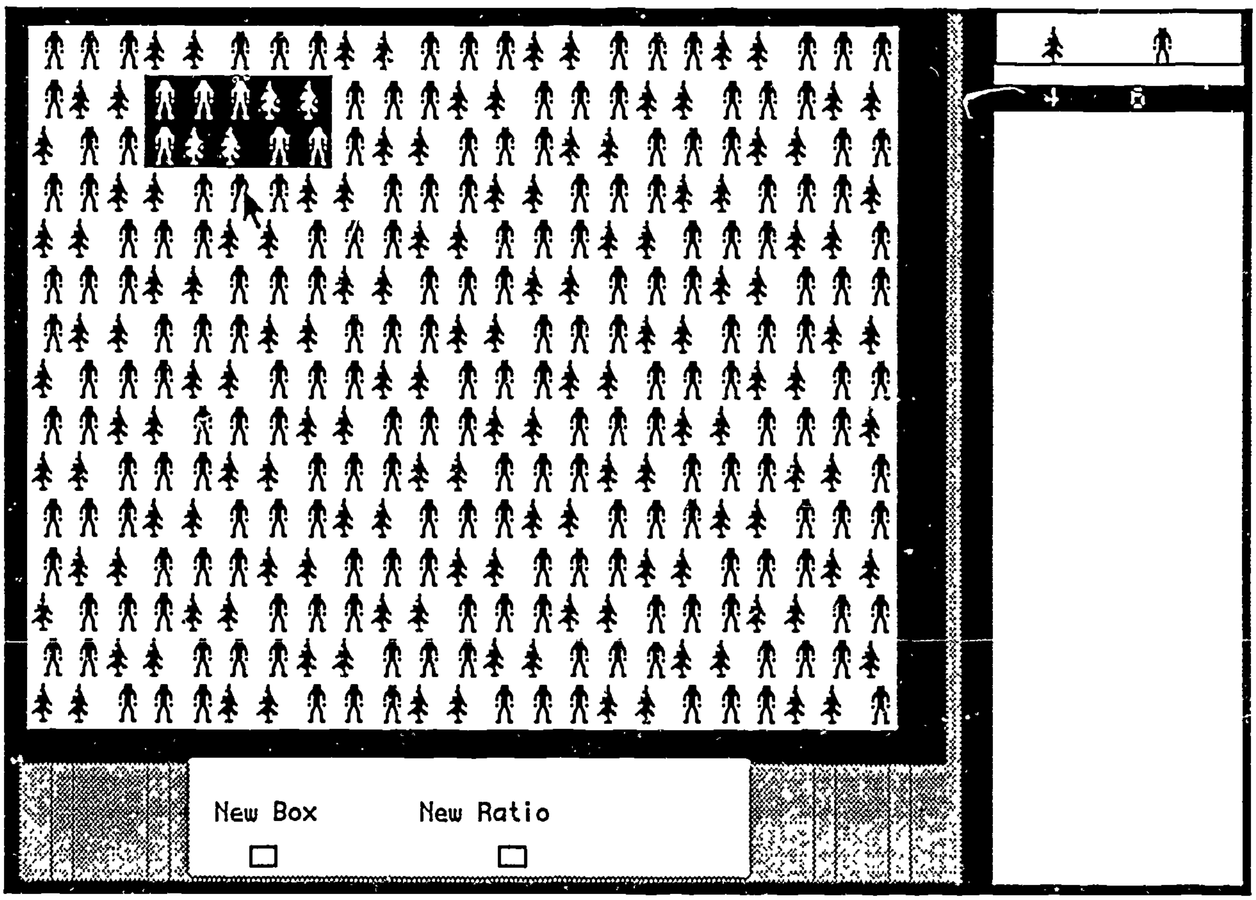

Figure 29

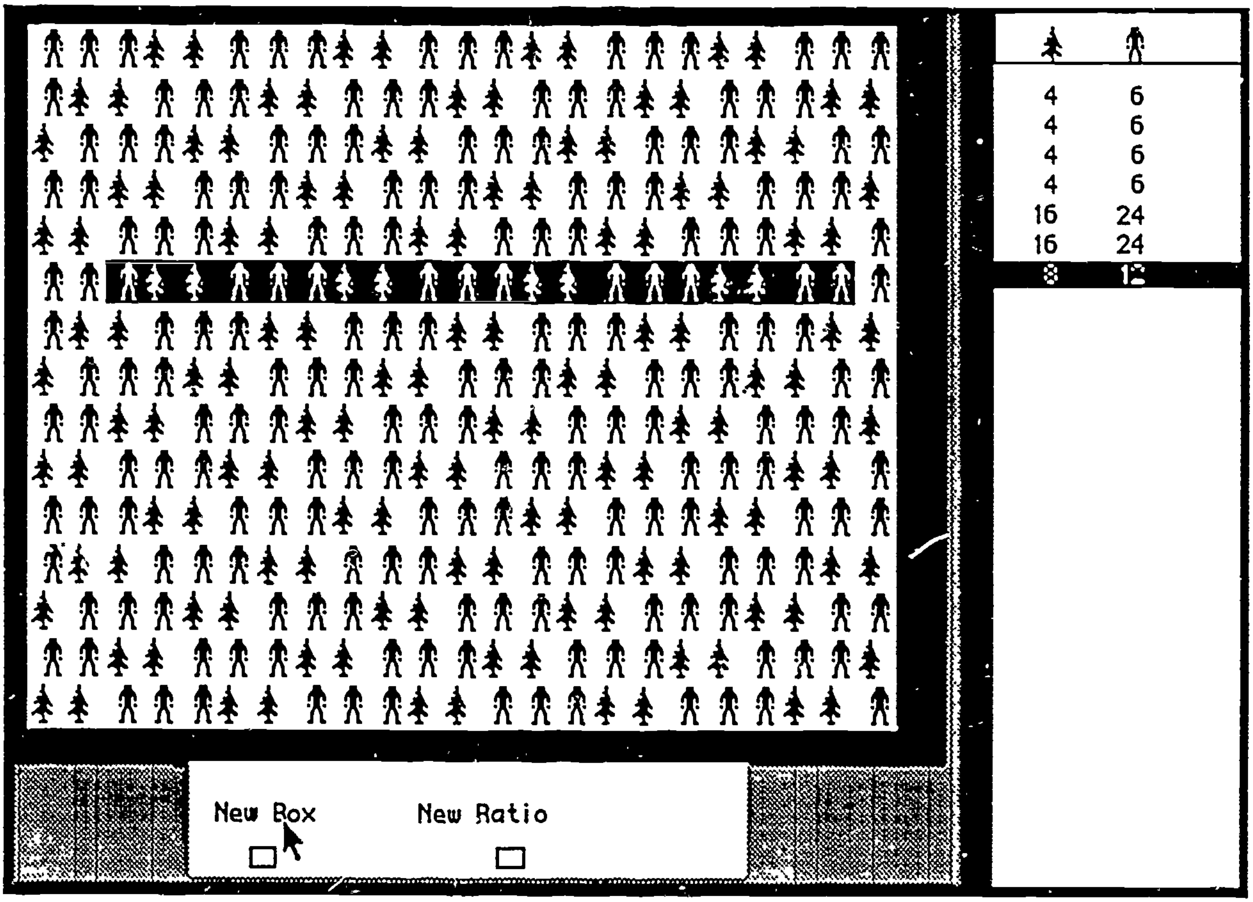

Figure 30

In Figure 30, we see the results of dragging this rectangle about to four different positions (each of which leads to the pair (4, 6)). We then used the new box command twice to generate new samples, all of which are recorded until the "new ratio" command is invoked, which takes the user back to a screen like that in Figure 27, but without the row of icons that represents the user's choice of ratio.

Another version of this environment supports the taking of arbitrary samples of a computer-provided ratio which the user then is to infer from samples taken, as in Figure 31. In Figure 31, one rectangle yielded the (1, 5) pair, and then a second rectangle was slid around to four different positions yielding different (non-constant) pairs, which indicates that the sample size is not adequate. The aim here is to determine the smallest homogeneous sample.

Another environment provides a coordinate graph of the data points of the sample in addition to the table, and yet another allows one to sample two different tesselations while comparing the respective data on a coordinate graph. We provide in Figure 32 a screen from the lat environment, where a homogeneous sample is guaranteed.

An option in this environment allows one to sample and compare unknown intensive quantities, where the sampling process is unconstrained. With each sample, a point is deposited on the coordinate graph and its coordinates are highlighted in the table.

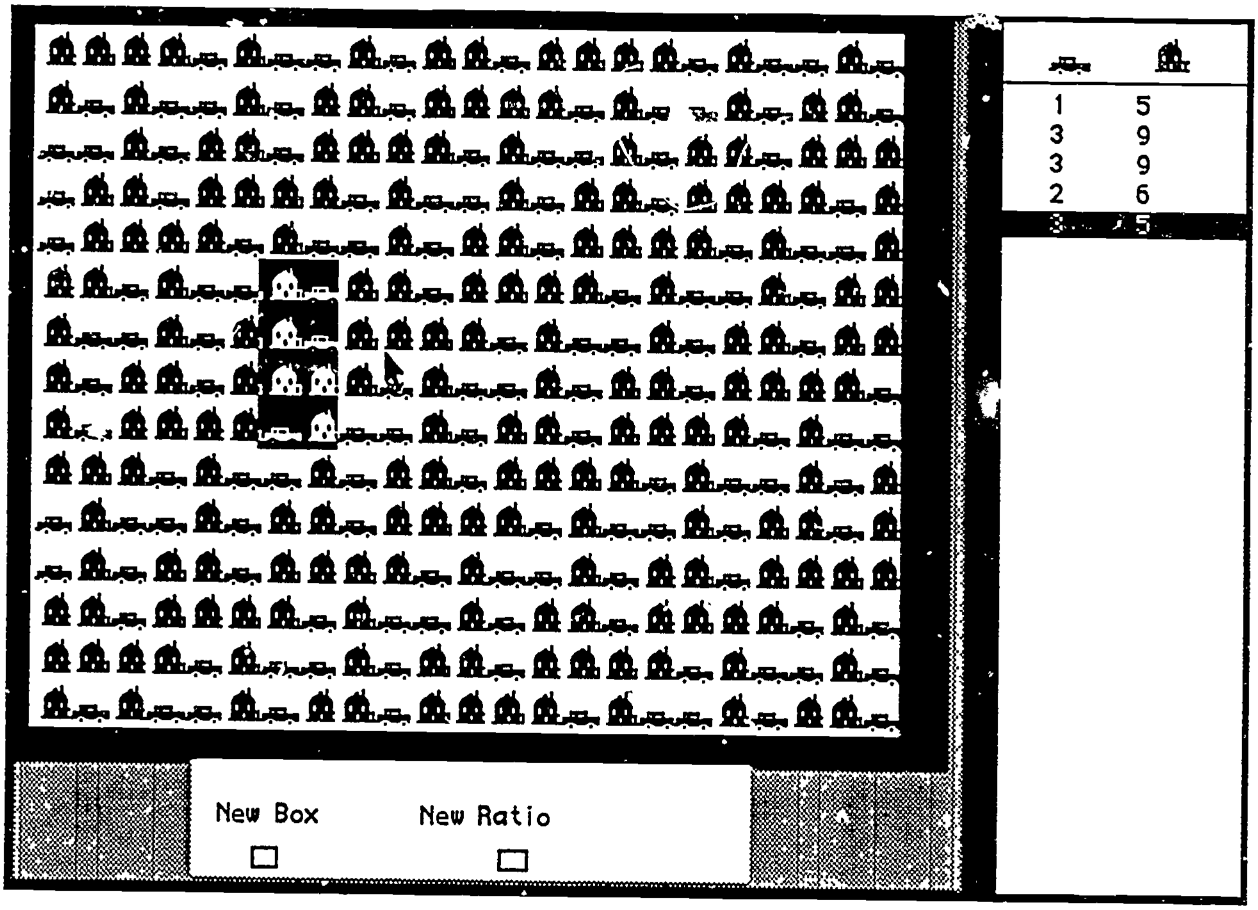

Figure 31

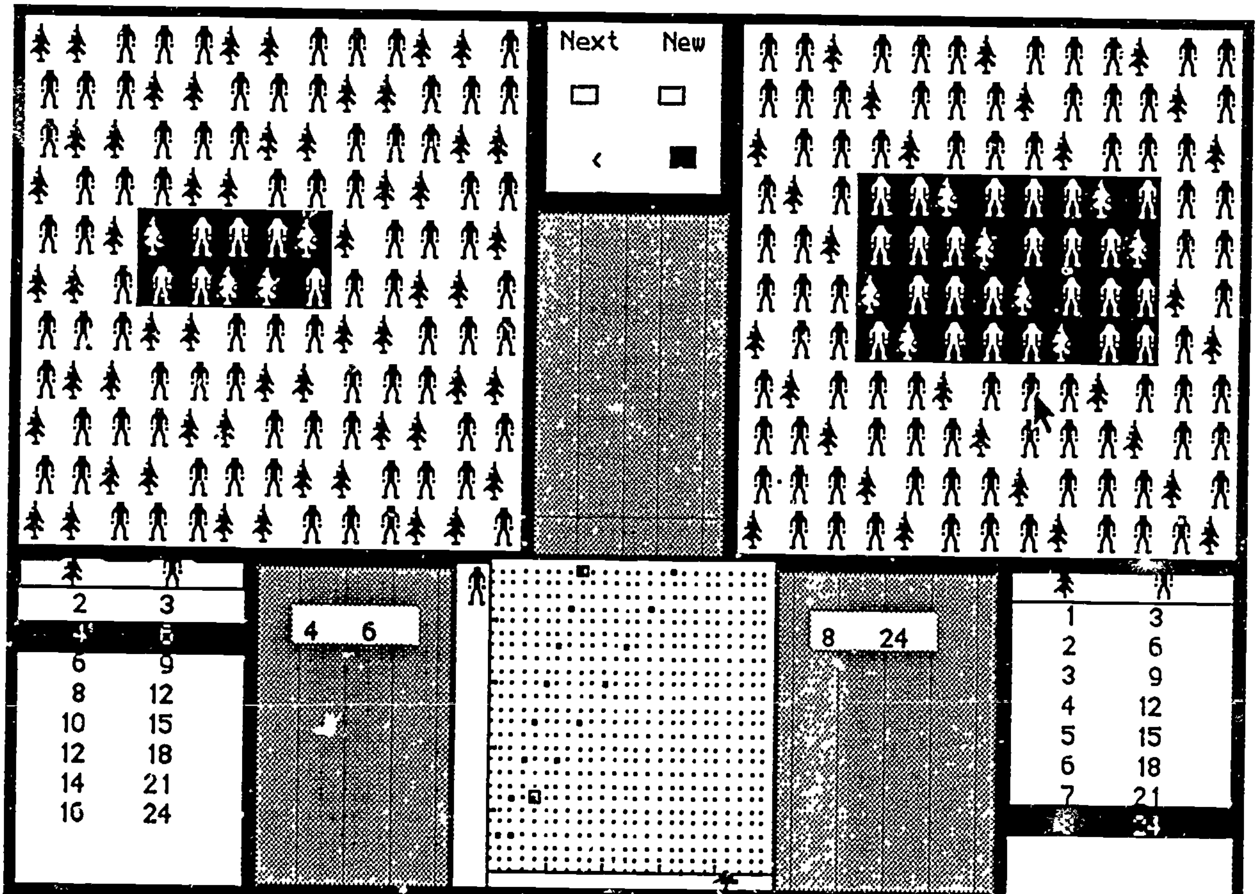

Figure 32

As is evident from this set of environments and certain of the earlier ones as well, we have available a set of representational components that can be assembled in a variety of ways to achieve a variety of objectives with a variety of student populations. In a different programming language, the options and windows that are fixed in position and size here could be made adjustable in both respects. Thus, looking at the previous example in Figure 32, one could expand the coordinate graph and shrink the tables to look at the distribution of samples on the coordinate graph, an especially interesting exercise when the samples are not homogeneous and the points do not lie on lines through the origin.

7. CONTINUOUS ENVIRONMENTS

Continuous Environments

- Transition From Discrete to Continuous
- Parallel Line Model
- Scaling Environment
- Linking Environment
  - Number-Pair Odometer
  - Coordinate Graph
  - Algebraic Equations

Chart 6

7.1 Introduction.

The bulk of the historically received mathematics curriculum beyond the first three or four grades concerns itself with continuous mathematics, presumably to build the mathematical knowledge needed to model the continuous aspects of quantitative experience. The conceptual issues and representations associated with continuous mathematics are considerably more complex than those of discrete arithmetic. It is for this reason that we chose to build conceptual structures and reasoning strategies in the world of discrete phenomena first. The aim is then to extend these to the continuous case using judiciously chosen transition representations and activities.

In the case of the traditional tabular and coordinate graphical representations, the extension is straightforward. In the table, we must decide on a step size and level of accuracy, but by its nature, the table is discrete. On the other hand, the coordinate graph is more naturally continuous because of the continuity afforded by the axes. Hence the extension in this case amounts simply to filling in points that were already available. In algebraic representations the issues are much the same as for the table of data: when incrementing, what step size to use; and when displaying, what accuracy to use. For a more detailed discussion of the features of these representations and how they match and fail to match, see (Kaput, in press).

The major issue at hand in the transition from the discrete worlds previously constructed to the yet to be implemented continuous environment concerns what should replace the iconic representation. Our choice is to introduce the continuous by using pairs of parallel lines and by using a command structure analogous to that used in the discrete environments.

The transition environments have been implemented in rough form and will be described below. The continuous environments have been designed but will not be implemented/tested until the last quarter of 1987. We will describe our designs below.

7.2 Transition From Discrete to Continuous Representations.

We have several different environments intended to provide a cognitive transition to the continuous environments, all based on the discrete Linking Environment. In particular, since we will be replacing the field of discrete object icons by a pair of parallel lines, we provide a transition to the parallel lines in much the same style as we provided transition to the coordinate graph in Section 4.3. We support incrementing and decrementing in exactly the same style as before - by means of MORE and FEWER buttons.

In Figure 33, we display perhaps the most elementary transition environment, where the <u>only</u> change from the standard discrete Linking Environment is the replacement of the coordinate graph with the parallel lines on which icons appear. (Note that these transition environments are modeling discrete phenomena. We are merely introducing the representational <u>forms</u> to be used in the continuous case.) In this figure, we see the results of 3 clicks on the MORE button - each click moved the "sweep line" to the right and highlighted (in inverse video) 2 tree icons and 3 people icons. Note the default scaling of the two parallel axes.

In Figure 34 we have replaced the table with a coordinate graph, but have maintained the icon window. Again, we see the results of 3 clicks on MORE.

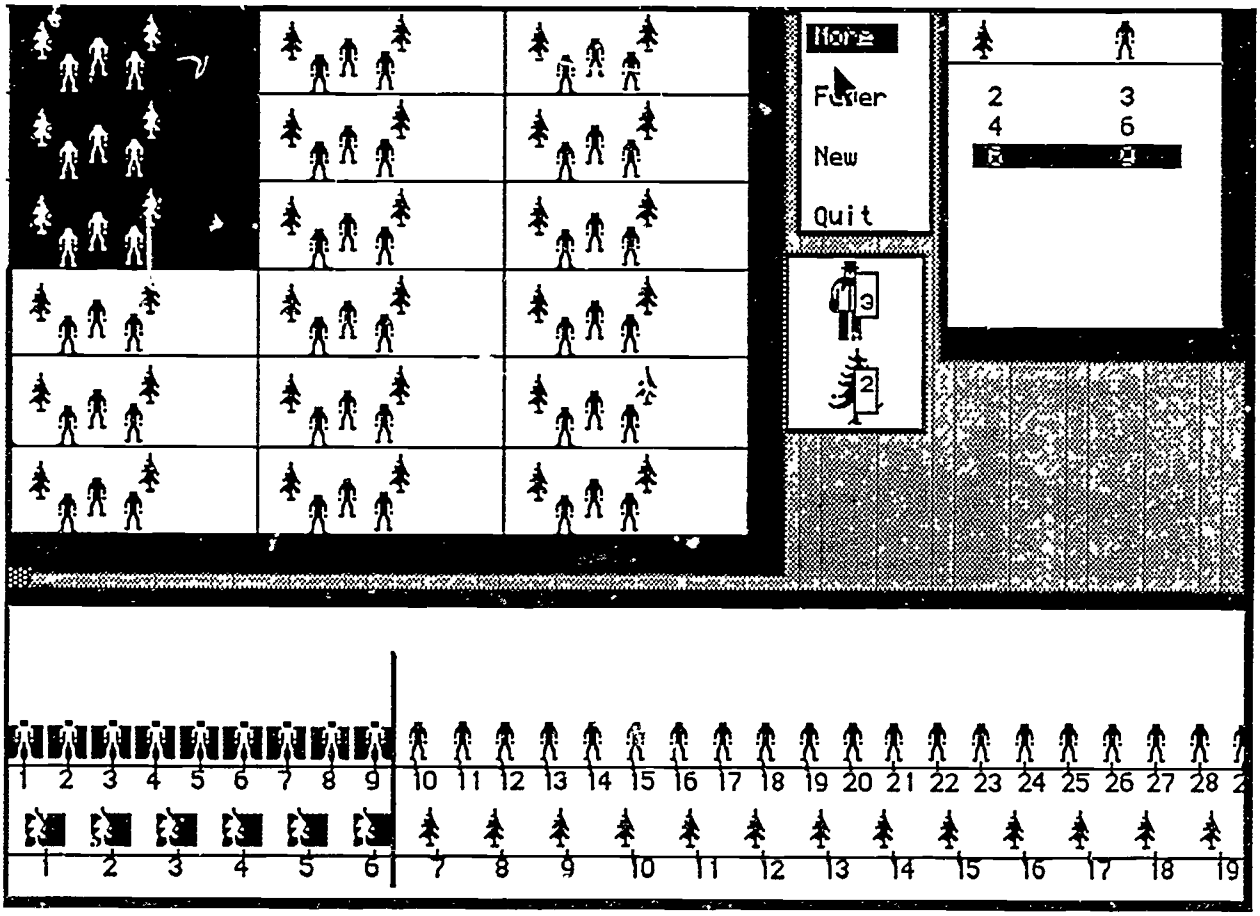

Figure 33

In figure 35, we have replaced the icon window by the coordinate graph and clicked on FEWER twice after having clicked on MORE several times.

As seen in the Sampling Envirnoment, we can mix and vary the representations at will while preserving their linkages.

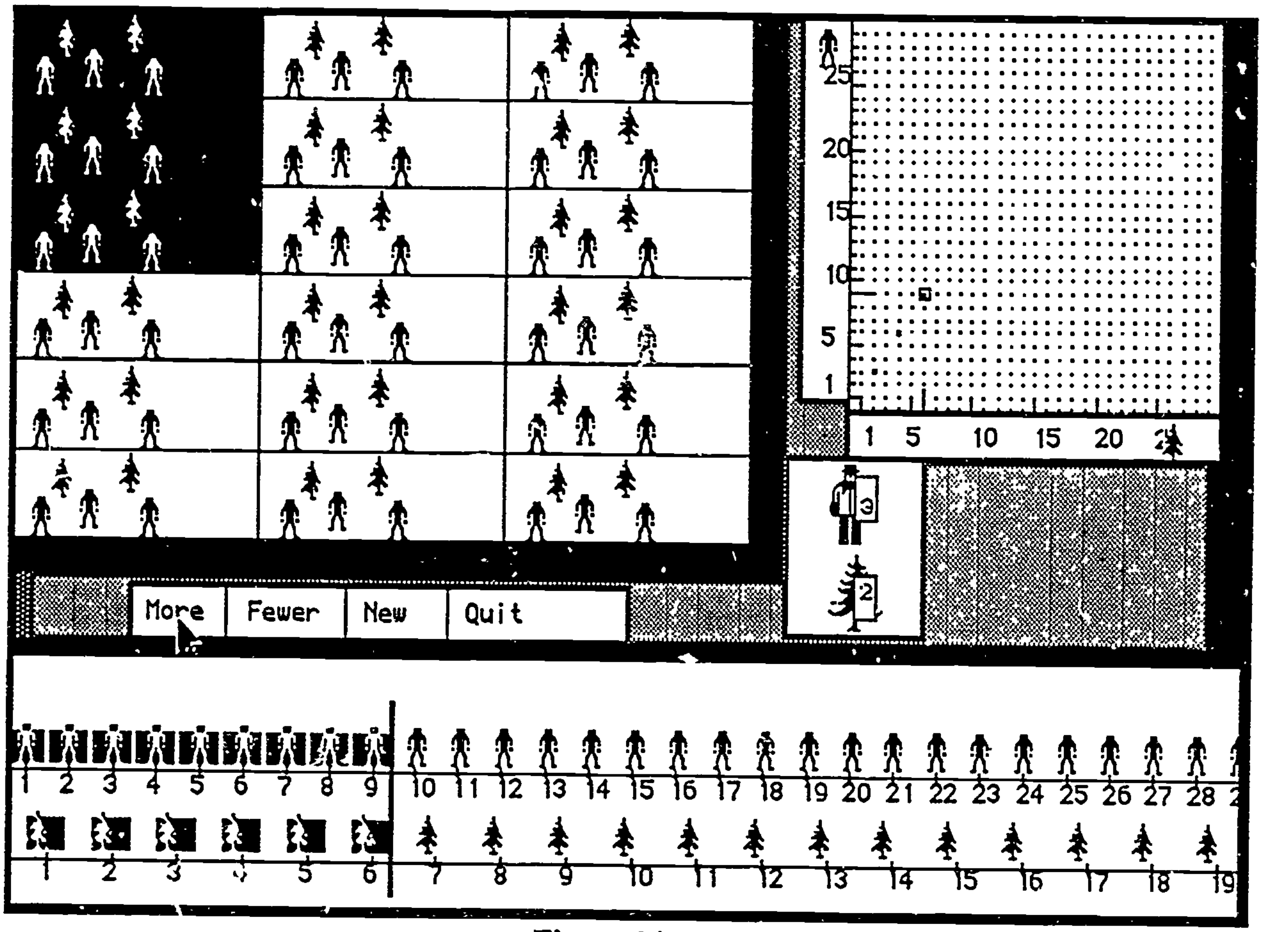

Figure 34

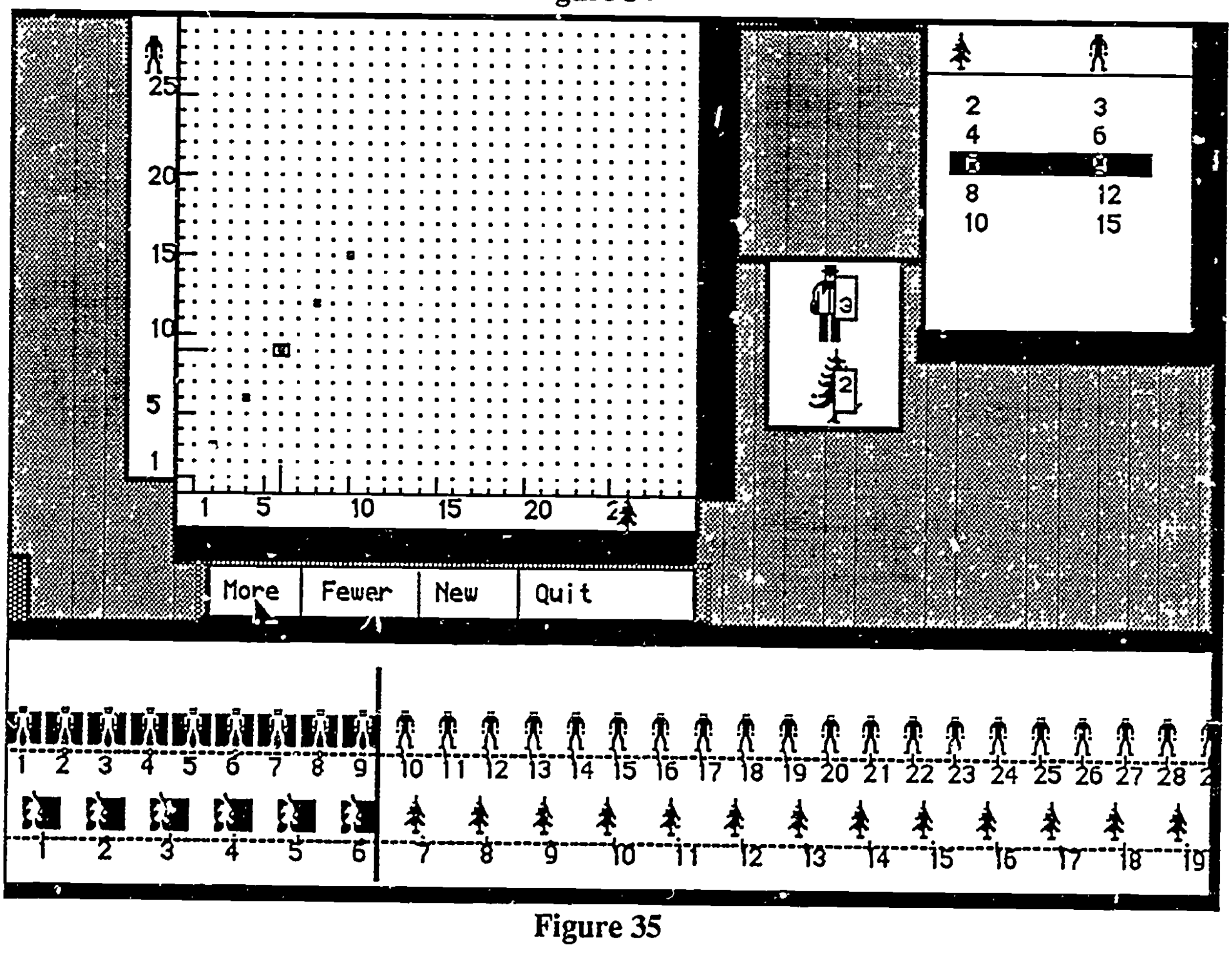

Figure 35

8. PLANNED CONTINUOUS ENVIRONMENTS

8.1 Introduction.

Our intention to complete the extension of the discrete software environments to the continuous case is based on the extreme importance of the wide curricular implications of this kind of software learning environment. It will provide concrete guidance for the type of large scale K-8 mathematics curriculum reorganization, including considerable compression and integration of algebra and arithmetic, that has recently been suggested in response to the devastating results of the Second International Comparison Study (McKnight, et al, 1987). Our research and development plans with respect to this portion of the software are much more modest than were our R&D activities associated with the development of the discrete environments. In particular, we will concentrate on the general demonstration of the import of the technology rather than on detailed cognitive research associated with its long term use. (This latter should, of course, follow.)

As already noted, we have dealt with discrete quantity first because its conceptual underpinnings seem simpler - acts of grouping and counting are simpler than acts of measuring, and the arithmetical representation of those grouping and counting acts is likewise simpler. Whole numbers are easier to deal with than rational numbers. Moreover, we have reason to believe, on the basis of data collected during the past year, that conceptual structures developed to reason with discrete quantity can be elaborated to reason with continuous quantity - transfer can be relatively direct. Lastly, by adhering to the modeling of discrete intensive quantities, we were able to maintain a distance from the competing conceptual structures associated with the part-whole aspects of rational number.

8.2 The Parallel Lines Model.

We have chosen to use a minimal model based on a pair of number lines to model a continuous intensive quantity, one for each of the two parts of the intensive quantity, e.g., one for distance (say, miles) and one for time (say, hours). Since this environment will be introduced after the student has experience with the discrete environments, we have an unusual opportunity to study the exploitation of a common command structure and screen layout in support of transfer, e.g., by simply replacing the MORE & FEWER commands by MORE and LESS commands. Holding the mouse button down on the MORE command will cause a line to sweep across the two parallel lines such that the two points at its intersection with the parallel lines to slide simultaneously along the respective lines leaving thickened trails behind them instead of leaving inverse video icons as in the transition environment.

As in Figure 36 below (a pictorial approximation to the parallel line window), if the lower line represents hours, the upper one miles, and the given intensive quantity represents, say a tractor mowing grass along a road at the rate of two miles for every three hours, then the upper point crosses the point corresponding to 2 (miles) when the lower one crosses 3 (hours), the upper crosses 6 when the lower crosses 9, etc. Under default circumstances, the two axes will be scaled identically, and number pairs will be entered in the table of data whenever the lower point crosses an integer.

One can make either or both lines display-active in response to the MORE and LESS or dragging commands, so that one can pose prediction problems - asking a student to predict the location of one point given that of the other (with the table of data turned off, of course).

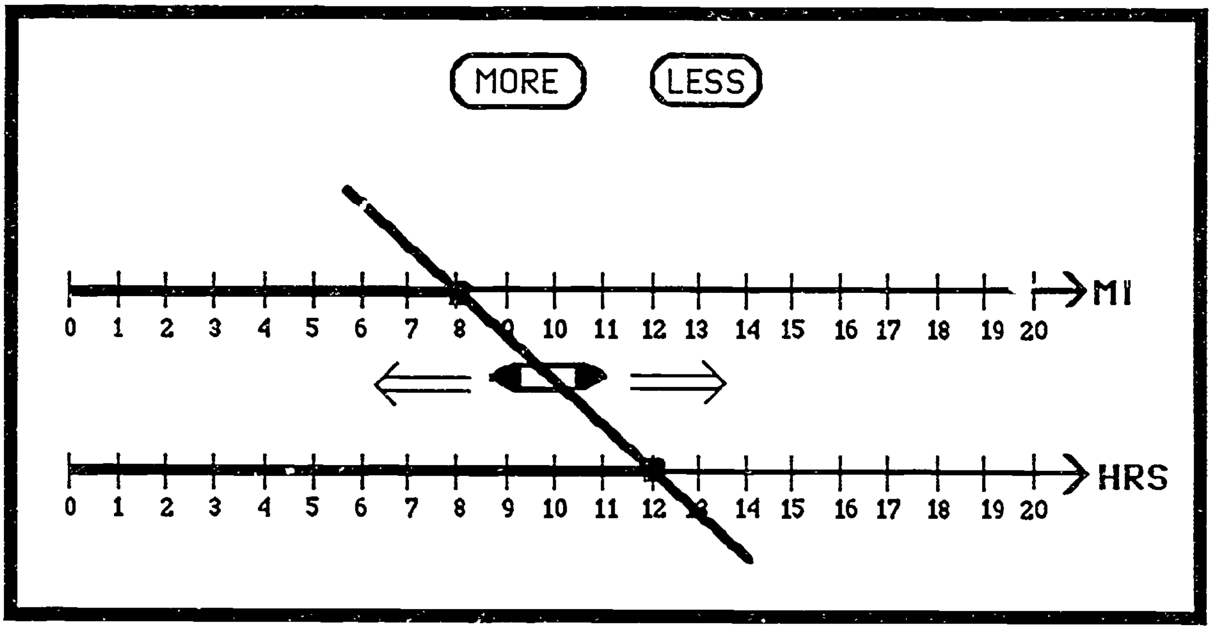

Figure 36

One can also drive the representation by dragging the sweep line as well. One points to either end of the "handle" on the sweep line between the axes, and, with the mouse button depressed, drags left or right. Furthermore, one can also enter pairs of numbers in the table of data by pointing to and clicking on any point on either line.

The auxiliary sweep line is intended to serve an important role - to provide an explicit feature to unite the two quantities being modeled on the respective lines into a single intensive quantity. This corresponds to the conceptual-support role of the rectangular cells in the discrete environments.

In support of the transition to the continuous world, and after introducing the transition environments of the previous section, we will enable the student to declare whether the quantities constituting the intensive quantity being modeled are discrete or continuous. The system then responds to the sweeping action by depositing either discretely placed points or continuous line segments, as appropriate, with an optional auditory beep signal if discrete quantities are involved. We should perhaps mention that many of the earlier discrete environments support a similar sound option to reinforce various counting or grouping acts.

For example, one might model a situation such as planning for a party where we assume that every two children drink three bottles of soda. In this case, both parts of the intensive quantity "bottles/child" are discrete, so the system would only deposit points at integral values, as indicated in Figure 37. (Note, however, that we would expect that only one "sweep" line would appear - the others are offered to indicate that multiple positions of the line are possible as one drags it about.) Moreover, we may allow students to use icons in the discrete case, as in Figure 37.

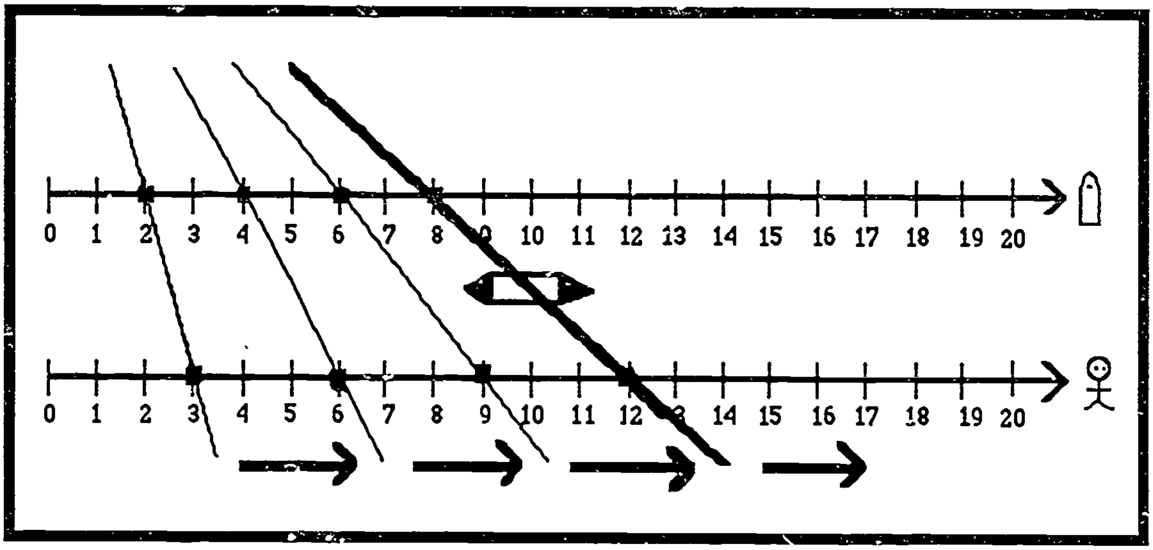

Figure 37

Situations exist that are mixed discrete and continuous, such as would occur if we chose to measure the amount of soda in terms of liters instead of bottles. Now, as pictured in Figure 38 below, the soda axis is measured in liters and is continuous.

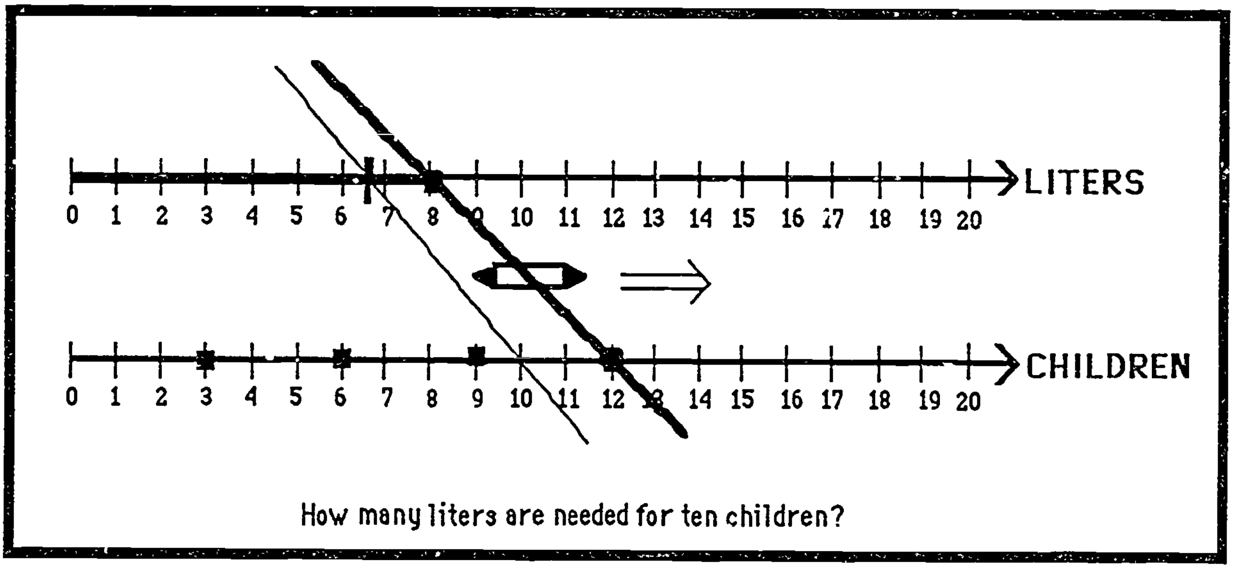

Figure 38

Further, one can then address questions such as "how much soda is needed if 1C children attend the party?" (See also the discussion below.) Depending on the mode in which the system is being used and which option is chosen, the system could display the response as a decimal in the table of data.

8.3 Scaling and Rescaling.

Scaling and rescaling will be a direct and easily accomplished matter: The student points to and clicks on the place where the unit will be (and tic marks are immediately generated on the remainder of the line), and then the student types in a number at that point to indicate what the unit magnitude will be (and the remaining tic marks are appropriately labeled). The default, of course, is 1. The process, when repeated (i.e., when it is a rescaling), leaves a "shadow of its former self" for comparison purposes adjacent to the new the newly scaled line. (This shadow can be made to

disappear as desired.) We feel that a rich set of scaling experiences is important at an early stage, so it is critical to provide easy and efficient scaling opportunities.

8.4 A Constructive Transition to the Coordinate Graph.

At the student's discretion, the upper of the two number lines (which consistently will be thought of as the "dependent variable" axis) can be dragged to a position orthogonal to the lower, meeting at their respective origins, thereby yielding the coordinate graph representation.

This constructive bridge to the coordinate graph representation is intended to tie students' conceptual structures associated with concrete understanding of those intensive quantities as descriptors of particular situations, especially those based on time as a parameter, to the more abstract coordinate graphical understanding of intensive quantities as slopes of straight lines. The experience of moving pairs of points in the parallel lines representation provides a simple model for continuous intensive quantities that shares features with many situations modeled by such quantities, in particular, constant rate of change situations, where time is the tacit independent variable.

Every scaling action in the parallel line configuration can be executed in the orthogonal configuration, and when both configurations are present, then the default condition is for a scaling action taken in one configuration to be reflected in the other configuration. By experimenting with a variety of scalings of a single concrete intensive quantity such as speed in the parallel line representation, where the model is simple and direct, we expect first that the student's cognitive representation of that intensive quantity will be enriched. Then, via the technique of examining analogous scalings in the coordinate graph, the student will extend that enriched understanding to the coordinate graph representation. Moreover, the prior experience with discrete intensive quantities is intended to provide a conceptual framework and conceptual entities to use to help encode the new phenomena being modeled.

As with the discrete world software, there will be an algebraic representation available - in the form of a set of equations that reflect inputs in other representations as well as inputs in their own variables (which, of course, will be reflected in the other representations on demand). The important difference here, of course, is that the variables can take on any rational values - not merely whole numbers. The default assumptions are that step size when variables are incremented, and the levels of accuracy in their algebraic display, will match those of the table of data.

8.5 Strategies and Reasoning Patterns In The Continuous Representations.

The easy-scaling feature described above allows one to rescale in such a way as to support certain important ratio reasoning patterns: For example, if every three children drink two liters of soda at a party, then one can answer questions such as

"How many liters of soda will be drunk by 10 children?"

using what might be called an "alias strategy."

Given a standard unit scaling of the (upper) children line, the student can scale the liters-of-soda line to show a tic mark at each even number directly below the multiples of three. Then the sweep line will be vertical and will move parallel to itself. The number on the liters-of-soda-line that corresponds to 10 on the children-line, is the answer, as in Figure 39.

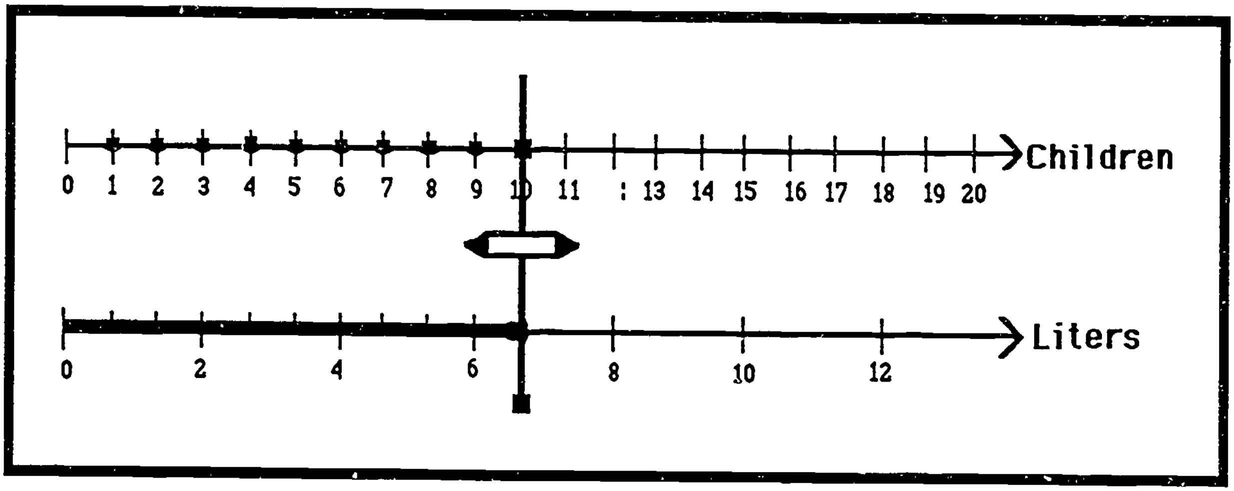

Figure 39

Note that we deposit a half-tic on the **Liters** axis whenever a whole number of children is swept out, and these conspicuously mark each liter interval of length two into thirds. Hence the answer can be recognized as represented by the point one third the distance from 6 to 8. (We should point out that, as long as we do not attempt to construct a coordinate graph from the lines, we can put them in any order we wish.)

Most especially, one can also ask how many bottles of soda are needed to serve 1 child - the number corresponding to one third of 2. Indeed, this is an ideal environment in which to introduce unit ratio strategies.

We refer to this scaling-based strategy as the "alias" strategy because tic marks at the same position have different names on the respective lines. It can also be regarded as a continuous line version of the "boxes" strategy - where boxes correspond to paired line segments (of length 3 on the upper line and length 2 on the lower line) associated via the vertical alignment of the tic marks representing values at multiples of 2 and 3, respectively. The successive computations of division and multiplication are embedded in the sweeping of the line, and the coordination of the two is provided by the vertical alignment and the automatic linear ordering of the values of each variable. (In particular, looking more closely at how the existence of the rectangular cells supports that strategy, we see that the availability of a certain <u>single</u> set of boxes to carry the numerosity of <u>both</u> sets of icons is at the heart of the strategy: the number of icons in one icon set leads to the number of boxes, which then, because the number of the other icons per box is attached to that box, determines the total number of icons in the other set. This attachment is accomplished by the sweeping line.)

Of particular, indeed <u>critical</u>, interest will be the ease of transfer of the boxes strategy to this new representation. This will provide not only information about the new software environment, but will provide important information about the quality of the cognitive structures built during experience in the discrete environments.

9. CONCLUDING REMARKS.

9.1 What We Have Done: Cybernetic Manipulatives.

The experience of building and testing the environments described in the body of this report has helped clarify certain powers of the computer as a representational medium. We now recognize more clearly how different representation systems support different forms of learning and different cognitions, and the potential for linking these in the student's mind. Further, we have clarified the

appropriate scale on which to approach the examination of the relationship between the mathematics curriculum and information technology.

With respect to curriculum, the appropriate unit of analysis and development is the conceptual field (Vergnaud, 1983) rather than particular topics, grade levels, or even "topic strands." The conceptual field we have dealt with here is the field of multiplicative structures. Such a conceptual field develops over a period of several years and needs a coherent curricular approach. In our case, for example, the handling of multiplication and division should be consistent with the idea of linear relationship, which is the mathematical context for ratio and intensive quantity.

Enlargening the curricular unit of analysis to the conceptual field implies enlarging the unit of learning environment design and application, especially with respect to the set of representations appropriate to that conceptual field, but also including a consistent interface and pedagogical style. Issues of fitting and actively linking representations become paramount.

Our ideals have been identified, but not fully achieved by the prototypes described earlier. A different development system and considerably more time would be required to accomplish this. Nonetheless, we feel that we have achieved student-intelligible, explicit links between concrete and abstract representations which help form a smooth ramp upward from the concrete to the abstract. We have done this in a pedagogical style whereby movement on that ramp is accomplished through student-initiated actions on those representations, and where the students themselves inspect the consequences of their actions, frequently in other representations that they deem appropriate.

Moreover, the abstract representations introduced here support the doing of mathematical thinking on quantitative relationships which are not linear. By design, our approach is intended to support generalization to the wide body of mathematics that lies beyond ratio and proportion - that is, beyond linearity.

In addition to curriculum/technology implications, there are important pedagogical implications.

At the lower end of the representational ramp, we have demonstrated the use of the computational medium to instantiate cybernetic manipulatives. Despite the widely acknowledged value of concrete manipulatives, they are not widely used in schools for two reasons:

(1) They impose a difficult classroom management problem. The fact that manipulatives are so rarely used and used successfully despite many years of promotion and positive research findings indicates the height of this management barrier. And of course, they are virtually never used in mathematics classes beyond the earliest grades.

(2) They impose a difficult cognitive management problem. It is difficult to use them in ways that adequately expose the connections between actions on the manipulatives and the corresponding actions on their formal mathematical representations, especially connections between actions on concrete models and the formal representations that constitute the language of mathematics. The cognitive load in managing the connections between the actions while performing the actions themselves is too great.

Environments of the type we have been describing help solve both of these problems. But in addition to providing the control and regularity that screen objects vs physical objects provide, thereby suitably simplifying the actions taken, the computer is able to expose the needed link between the concrete and the more formal representations in a time-independent way, i.e., displayed at the user's discretion. This provides enormous pedagogical opportunity for discussion, cognitive conflict resolution, prediction, and so on.

9.2 What We Have Not Done.

We have not addressed the matter of building learning environments for the broader set of constructs known as "rational number." In particular, we have not dealt with part-whole relations or with the arithmetic of fractions. It remains to be seen how much of the approach taken here

or with the arithmetic of fractions. It remains to be seen how much of the approach taken here extends or is consistent with natural approaches to these other rational number constructs.

More importantly, we have not described the curricular materials to be used in conjunction with the learning environments we have constructed. In the course of building and testing these environments we have produced a variety of written materials and procedures which will be described elsewhere. We regard these materials, and the ways that they are used, as at least as important as the software environments themselves.

REFERENCES

Behr, M., Lesh, R., Post, T., & Silver, E. (1983) Rational number concepts. In R. Lesh & M. Landau (Eds.) Acquisition of Mathematics Concepts and Processes. New York: Academic Press.

Kaput, J. (1985) Multiplicative word problems and intensive quantities: An integrated software response. Technical Report 85-19. Educational Technology Center, Harvard Graduate School of Education, Cambridge, MA 02138.

Kaput, J. (1987) The cognitive foundations of models that use intensive quantities. Paper prepared for the Third International Conference on the Teaching of Modelling and Applications. Werner Blum, Chairman. Kassel, Federal Republic of Germany.

Kaput, J. (in press) Representation in the symbol systems of algebra. In C. Kieran & S. Wagner (Eds.) A Research Agenda for Algebra, (Research Advisory Committee Monograph Series) Reston, VA: National Council of Teachers of Mathematics.

Kaput, J., Luke, C., Poholsky, J., & Sayer, A. (1986) The role of representations in reasoning with intensive quantities. Technical Report 86-9. Educational Technology Center, Harvard Graduate School of Education, Cambridge, MA 02138.

Kaput, J., Luke, C., Poholsky, J., & Sayer, A. (1987) Multiple representations and reasoning with intensive quantities. Proceedings of the 11th Annual Meeting of the PME-NA, J. Bergeron, N. Herscovics, C. Kieran (Eds.) Montreal: University of Montreal.

McKnight, C., F. J. Crosswhite, J. A. Dossey, E. Kifer, J. O. Swafford, K. Travers, T. J. Cooney (1987) The Underachieving Curriculum: Assessing U.S. School Mathematics from an International Perspective. Champaign, IL: Stipes Publishing Co.

Schwartz, J. (in press) Extensive and intensive quantities: Referent preserving and referent transforming quantities. In M. Behr & J. Hiebert (Eds.) Research Agenda Project: Number in the middle grades, (Research Advisory Committee Monograph Series) Reston, VA: National Council of Teachers of Mathematics.